2.11 DECISION MAKING ELEMENTS

The C3I process performs the decision making that controls the activities of the other three EADSIM processes, Detection, Flight Processing and Propagation. A platform’s assigned ruleset determines the specific actions and reactions of the platform in carrying out its mission. Using rulesets, the C3I decision making process governs the interaction between participants within a scenario. Several factors influence these interactions: command relationships, scripted targets, the dynamic air picture as perceived by the platform, and weapons on the platform.

While there are a number of different rulesets in EADSIM, the processing performed by each consists of three areas:

- Battle Management/C3 (BM/C3)
- Message Processing
- Track Processing

The BM/C3 and Message Processing components of the C3I process are highly interdependent. Message Processing determines the response at a given platform to messages received from other platforms. These messages in turn influence activities performed by the BM/C3 phases. They may also be forwarded or may generate additional messages depending on the Message Processing function of the ruleset. BM/C3 phases model the participant's decision and engagement processes each of which determine the time required to make an assignment decision or to perform an engagement.

Track Processing for each ruleset is independent of BM/C3 and Message Processing phases. The Track Processing associated with a ruleset determines how track data is handled on a platform.

Battle Management and Message Processing: Battle management/engagement decision processing, as well as the engagement modeling, for the simulation is contained in the ruleset phase and message processing. Rulesets are grouped by the categories shown in Figure 2.11-1. These groupings indicate the general type of battlefield participant that the ruleset was designed to model.

AIRBASE

AIRCRAFT

- BOMBER
- AIR-GROUND ATTACKER
- RETURN TO BASE
- WAS BOMBER
- AIR REFUEL TANKER
- BOMBER
- FIGHTER BOMBER
- ESCORT
- SWEEPER
- WILD WEASEL

AIR COMMANDER

- FLEXIBLE COMMANDER
- GROUND ATTACK COMMANDER

DIRECTED ENERGY WEAPON

- LASER
- FLEXIBLE COMMANDER

GENERAL

- DELAY
- GENERIC
- NO COMMAND
- WEAPON EVENT

SAMS

- FLEXIBLE COMMANDER
- FLEXIBLE SAM
- MISSILE
- SAM LAUNCHER
- SAM LCS
- EMITTER DECOY

SENSOR PLATFORMS

- AWACS
- FLEXIBLE COMMANDER
- FLEXIBLE SAM
- FUSION POST
- GSM

SURFACE-TO-SURFACE CHAIN

- GROUND ATTACK COMMANDER
- SSM COMMANDER
- BTRY
- CARTY
- CTOC
- INTEL CAC
- MBNFDC
- RED TEL
- SS FU
- HIDE SITE
- LAUNCH SITE
- RELOAD SITE

FIGURE 2.11-1. Overview of Ruleset Grouping.

The rulesets used to build SAM command chains are the Flexible SAM, Flexible Commander, SAM Launcher Control Station (LCS), and SAM Launcher. The Flexible SAM ruleset is capable of modeling a wide variety of SAM systems and can be used with the SAM Launcher to model SAM systems with remoted launchers. The SAM LCS ruleset is used in conjunction with the SAM Launchers to provide access by multiple Flexible SAM platforms to groups of launchers. The Missile ruleset can optionally be used in conjunction with the Flexible SAM and SAM Launcher rulesets to provide for explicitly flown interceptors. Emitter Decoy rulesets can be used in conjunction with Flexible SAMs and Flexible Commanders operating to represent dynamic emissions control.

The Flexible Commander ruleset allows the user to model the range of command chains from complete decentralization of the firing platoon to the level of centralization at which both SAM and DCA aircraft are controlled by a common commander. The Flexible Commander ruleset is capable of commanding aircraft and can vector aircraft to ABTs. The Flexible Commander can also request engagements by SAM systems not directly under its command. These can include engagements against ABTs and tactical missile threats, with optional shared control of the interceptor flyout against those targets.

The sensor platform rulesets allow modeling of platforms whose function is to provide track data. These include the AWACS, GSM, Fusion Post, the Flexible SAM, and the Flexible Commander rulesets. The GSM was designed to model the JSTARS Ground Support Module. The AWACS ruleset provides the capability to report data, taken from a platform's local sensors, across communications links. The Fusion Post ruleset represents nodes that can collect data from multiple sources including its own sensor and report a recognized air picture out over one or more connections. The Flexible SAM ruleset is considered as part of this category, since the functionality represented by the AWACS and Fusion Post rulesets is a subset of the functionality of many of the other rulesets (including the Flexible SAM) with the track reporting phase. The Flexible Commander is also a part of this list since it has the capability to provide track-updates for surface-to-air and air-to-air interceptors that are in flight to their targets.

There is a single Airbase ruleset that represents the functionality of the airbase in support of the air battle.

The Laser ruleset is used in modeling Directed Energy Weapons (DEWs). This ruleset can be used for ground-based, airborne, or satellite-based DEW platforms. The Flexible Commander ruleset can be used in conjunction with the Laser ruleset to provide many of the deconfliction and command options provided for the SAM systems.

Several of the rulesets are designed to operate on aircraft performing specific air missions. The Fighter ruleset represents aircraft in a DCA role. The Air-Ground Attacker (AGAttacker) ruleset represents aircraft that can perform scripted or commanded engagements against ground targets. It can also engage targets of opportunity. The AGAttacker can also defend itself using air-to-air weapons should it be attacked by enemy aircraft. The Was Bomber and Return to Base (RTB) rulesets are transitional rulesets used to represent the capabilities of other rulesets once their primary mission is either completed or interrupted. The Fighter can be commanded by the Flexible Commander, while the AGAttacker receives its commands from the Ground Attack Commander. The Air-Refuel Tanker ruleset represents aircraft that can provide airborne refueling to other aircraft.

Several of the aircraft rulesets have been subsumed by the capabilities available within the Fighter and AGAttacker rulesets. These rulesets are still available; however, their capabilities are significantly more limited than those available within the Fighter and AGAttacker. These limitations generally center around lack of communications and coordination of targets. The Bomber ruleset represents an aircraft on a bomber mission with no active defensive capabilities against air attack. A platform with an Escort ruleset can be used to provide air support to a platform on a bomber mission. The Fighter/Bomber ruleset is designed to provide for modeling of a combined air-to-surface and air-to-air capability with a primary mission against surface targets.

The Wild Weasel ruleset represents a set of tactics for a platform flying in the Wild Weasel role. The Bomber, Fighter/Bomber, and Wild Weasel are limited to specific scripted targets by name. The Sweeper ruleset is an aggressive, autonomous fighter with no communications and no deconfliction of engagements with anyone.

The surface-to-surface rulesets were designed to model artillery fire in both dynamic and scripted modes. The Ground Attack Commander provides command of both airborne attack and surface-to-surface attack of surface targets. The SSM Commander is a more limited case of the Ground Attack Commander providing command of the surface-to-surface weapons against targets. The Intelligence Collection and Analysis Center is used to collect and provide targeting information to the various ground attack components. The SS FU ruleset supports both scripted and dynamic launches of Tactical Ballistic Missiles

(TBMs), while the Red TEL provides the primary mechanism for scripted launch of TBMs. The various site type rulesets are used in conjunction with movement and concealment of the SS FU.

Battle Management Phases

The battle manager of each ruleset operates under a set of phases that determine the activities conducted by the platform at a given time. Table 2.11-1 lists the phases by ruleset in EADSIM. The Fighter ruleset phases are typical of the aircraft rulesets: Target Select, Vector, Engage, Lock, Launch, Intercept, React-to-Engage, React-to-Lock, Drag Maneuver and Jammer Control. The Target Select phase performs threat assessment and target assignment for the flight leader. During this phase a target list is established containing the targets the flight can engage. The target select phase is rescheduled periodically (i.e., at a fixed time interval) for the fighter until a target is selected. The Vector phase is scheduled for the flight leader when a target is found during Target Select or when a commanded assignment is received from the flight leader's commander. During the Vector phase the flight is diverted (the entire flight follows the flight leader) to an intercept point or to a specific target. The vector phase is rescheduled periodically until a target is to be engaged at which time the Engage phase is scheduled. In the Engage phase targets are assigned to individual fighters in the flight. Each fighter enters the Lock phase as it is assigned a target and remains there until it achieves lock on the target or 10 seconds have elapsed. Lock is achieved when the weapon - target geometry satisfies the weapons angle constraints. Once lock is achieved, the fighter enters into the Launch phase during which the fighter determines whether a launch occurs. Launch will occur if the weapon's angle constraints continue to be satisfied and the target is within the capable range of the weapon. If launch occurs, the fighter will enter the Intercept phase. When the range from the weapon to the target is zero the intercept is evaluated to determine whether a kill has occurred or the target should be reengaged. If the weapon had semiactive guidance, the fighter is revectored when the target is killed. Otherwise, the weapon is fire and forget and the fighter was revectored in the Launch phase. When a fighter enters the Engage phase, the Engage phase schedules the target to enter the React to Engage phase. In this phase the target will determine if and how it will react to the engagement. When the fighter enters the Lock phase, the Lock phase will schedule the target to enter the React to Lock phase. In the React to Lock phase the target will determine whether and how to react to the attacker's lock. The Drag Maneuver phase is scheduled by the React to Engage or the React to Lock phases and determines the target's response to the attacker at the end of the drag maneuver (options include engage, lock, return to base, etc.). In the Jammer Control phase The fighter can dynamically activate and point jammers in response to emissions detected by on board SIGINT sensors. It can also activate and deactivate scripted jammers.

TABLE 2.11-1. Ruleset Phases.

RULETYPE	PHASES	RULETYPE	PHASES
AIR REFUEL TANKER	React to Engage React to Lock Drag Jammer Control	FLEXIBLE COMMANDER	Target Select Jammer Control Counter Measures
AIRBASE	Target Select	AWACS	Jammer Control
CARTY	Target Select	CTOC	Target Select
FUSION POST	Target Select	FLEXIBLE SAM	Target Select Launch Intercept Reload Jammer Control Counter Measures
ESCORT	Lock Launch Intercept React to Engage React to Lock Drag Jammer Control		
BOMBER	Target Select Lock Launch Intercept React to Engage React to Lock Drag 1 Jammer Control	FIGHTER-BOMBER	Target Select Lock Launch Intercept React to Engage React to Lock Drag Jammer Control
FIGHTER	Target Select Vector Engage Lock Launch Intercept React to Engage React to Lock Drag Jammer Control	AGATTACKER	Target Select Lock Launch Intercept React to Engage React to Lock Drag React to SAM Lock Jammer Control

Track Processing

Most battlefield participants rely on track information from local and remote sources for situation awareness, battle management/engagement decisions, and weapon guidance. The storage and dissemination of this information is dependent on the source of the information and the type of participant being modeled; thus, it is dependent on the ruleset in the EADSIM.

Local sources are sensor sources that feed track data directly into a specific system or component of that system. The AN/APG-63 on an F-15 fighter is an example of a local track source. Where local track is required in the simulation, e.g. intercept with a semi-active missile, the sensor source must be on the platform requiring local track. This allows track data to be directly processed into the track file of the platform modeling an unjammable communications link to the sensor source for the fire control process.

Remote sources of track data are sensor, fire unit, or command and control (C2) nodes that provide track information across a communications link. Remote-source tracks are not usable in the fire control process where the modeled system must have track on a target for engagement purposes. This modeling of local and remote sources of data requires the co-location of the fire control radar with the engagement decision process, i.e. ruleset, on the system definition when the system is required to have local track. Assessment of partial damage as a result of a displaced fire control sensor for SAM systems is not possible. A SAM site is therefore either completely operational or dead.

Platforms store track information in three different track files; general track files, ground track files, or signal track files. The processing for all types of track files is similar. However, ground track files store data for ground tracks only. Signal track files store data only for tracks detected by a SIGINT or PASSIVE sensor. They are used to dynamically control jammers. The data SIGINT and PASSIVE sensors also feeds into the general track file or the ground track file for targeting decisions and dissemination.

Modeling key features of track files is essential to operations of rulesets that use track data. Track file management and data thread processing are identical for both general, ground, and signal track files. Track file management determines the number of tracks allowed in a track file and which tracks are entered or deleted in a track file. The source of the tracks determines operations performed. The source can either be a local source (i.e., a sensor on the platform, or a remote source) (i.e., a message from another platform). Dissemination of the tracks is also key to development of a complete recognized air picture.

Track file management is controlled by parameters specified in the platform’s ruleset. The two essential parameters that control track files are the track file size and the purge time. Track file size limits the number of objects that may be in track at any given time for both local and remote tracks. If this number is exceeded, a track may be purged to allow space for new tracks. A local track will be purged because of age if the local track purge time has expired since the last local track update. A remote track will be purged because of age if the remote track purge time has expired since the last remote track update. A track with both local and remote track data will only be purged if both the local and remote track purge time has expired. The purge time is used to determine the amount of time a track may remain in a track file without being updated. Additional parameters, such as AOI radius, track forwarding capabilities and AOI filtering influence the processing of the tracks, but are not specific to track purging.

The platform’s ruleset determines whether or not it has one of these files, which type of file it has, and the control parameters of that file. Table 2.11-2 lists the rulesets and the track files each can have. Note that several rulesets have no track files. These rulesets rely on other means for engagement and battle management decisions.

TABLE 2.11-2. Possible Track File Assignments by Ruleset.

Ruleset	**General Track File**	**Ground Track File**	**Signal Track File**
AGAttacker	x	x	x
Air Refuel Tanker			x
AWACS	x		x
Bomber			x
CArty	x		
Escort	x		x
Fighter	x		x
Fighter Bomber	x		x
Flexible Cmdr	x		x
Flexible SAM	x		x
Fusion Post	x		x
GA Commander	x		x
Laser	x		x
MBnFDC	x		x
Sweeper	x		x
SSM Cmdr			x
Was Bomber	x		x
Wild Weasel	x		x
CTOC			
SS FU			
Delay			
AirBase			
GSM			
Intel CAC			

2.11.1 Functional Element Design Requirements

Not currently available.

2.11.2 Functional Element Design Approach

Not currently available.

2.11.3 Functional Element Software Design

2.11.3.1 Flexible SAM Ruleset

Calling Tree

The calling tree for the Flexible SAM Ruleset consists of multiple trees corresponding to each of the 4 phases of the ruleset. The calling trees are shown in figures 2.11-2 through 2.11-5. Each tree begins with the main program for the phase which is invoked by the C3I process scheduler. Each phase is scheduled by another phase, either in the Flexible SAM or by another platform. The details of phase transition are discussed in section 2.1.4.4.1.

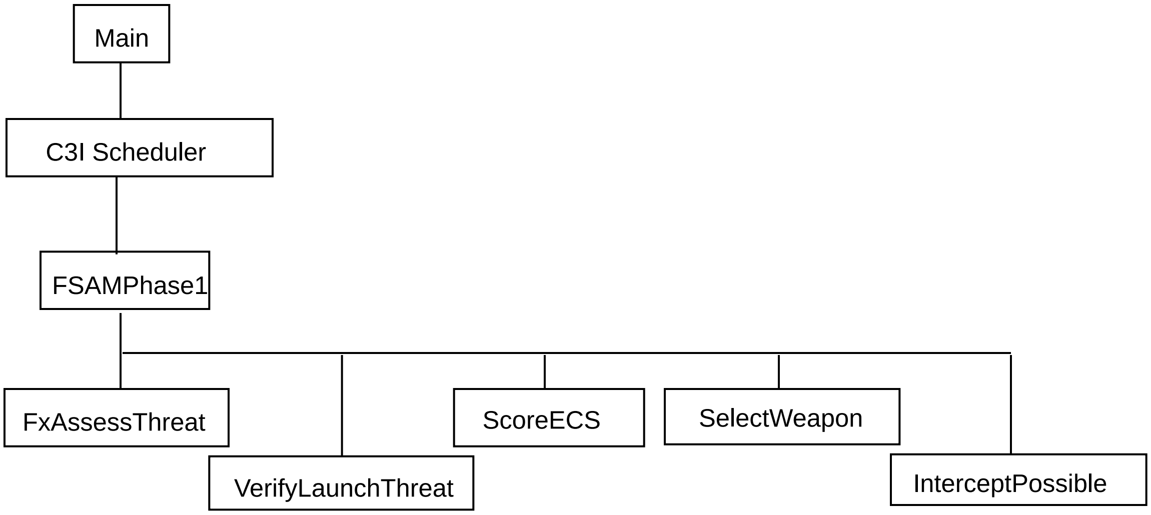

FIGURE 2.11-2. Calling Tree for Flexible SAM Phase 1 (Target Select).

Module Names and Descriptions for Phase 1

FSAMPhase1- is the target select phase for the Flexible SAM ruleset. FSAMPhase1 is responsible for assessing threats to particular assets, weapon selection and scheduling initial launches against targets.

FxAssessThreat- is called by FSAMPhase1 to assess threats to particular assets. It is responsible for assessing tracks, launch deconfliction, threat classification, and maintenance of the Asset Threat Category (ATC) List.

SelectWeapon- SelectWeapon is the module that by the Flexible SAM ruleset to select the best weapon to engage a target. SelectWeapon selects a weapon, computes the estimated time of intercept, the number of shots to be taken against target, the illumination time required by the weapon for the target, returns the ID of the IFTU platform that will track the intercept.

VerifyLaunchThreat - deletes pending launches against targets that the OpFac has not yet launched against and now no longer considered a threat.

InterceptPossible - Determines if an intercept starting from the given time is possible. Returns a true or false value.

ScoreECS - Provides a rough “score” for a Patriot ECS to aid in deconfliction of engagements at the battalion level.

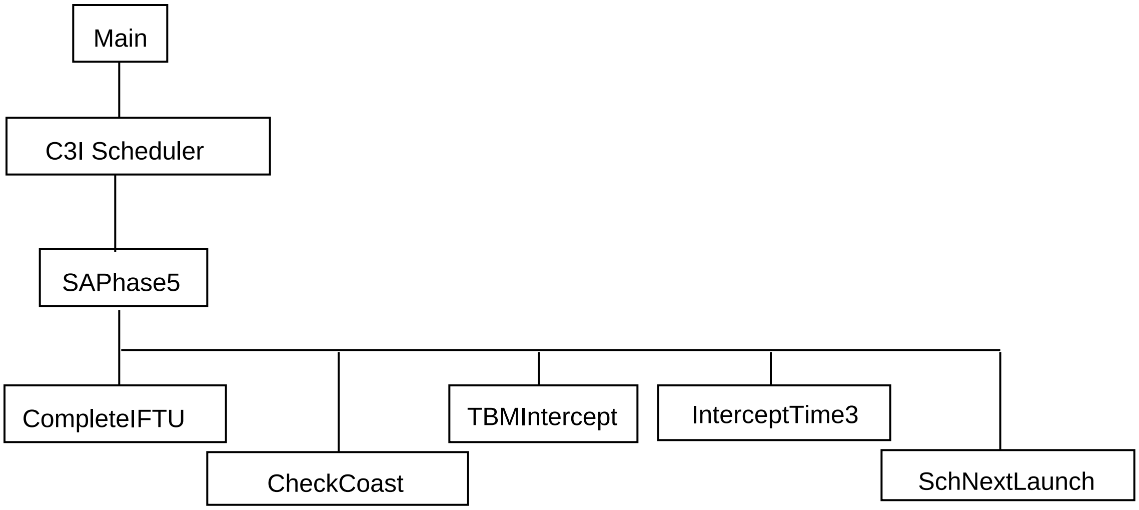

FIGURE 2.11-3. Calling Tree for Flexible SAM Phase 5 (Launch).

Module Names and Descriptions for Phase 5

SAPhase5- performs launch processing for the Flexible SAM ruleset. This module is scheduled to execute at the launch time of the ground based weapon. If the launch has not been canceled, the Flexible SAM will continue the engagement and coordinate any required remote launchers or in flight target update (IFTU) platforms.

TBMIntercept- is called by SAPhase5 to estimate the intercept time for a tactical ballistic missile (TBM) engagement. This module assumes the intercepting weapon is flying in a straight line at a constant velocity. It further assumes that the engagement control system and the launcher are co-located.

InterceptTime3- is called by SAPhase5 to estimate the intercept time for an air breathing threat (ABT) engagement. This module assumes the intercepting weapon is flying in a straight line a constant velocity.

CompleteIFTU - This is a utility function which determines whether an IFTU engagement should be terminated and notifies the launching platform that the engagement is complete.

CheckCoast - This is a utility function that determines if the allowed coast period for a target has been exceeded.

SchNextLaunch - SchNextLaunch is a simulation executive utility used to reschedule the ruleset launch phase.

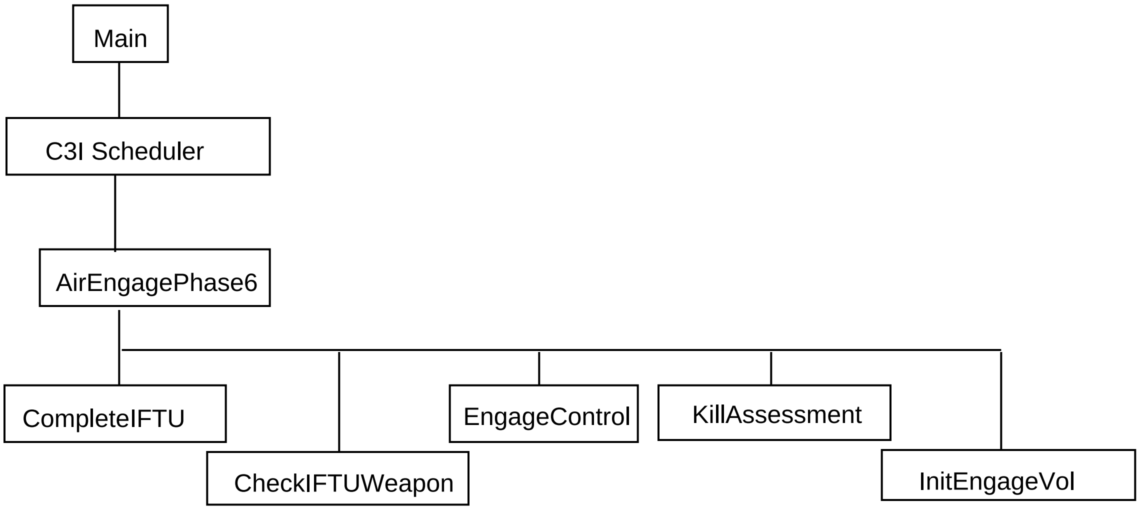

FIGURE 2.11-4. Calling Tree for Flexible SAM Phase 6 (Intercept).

Module Names and Descriptions for Phase 6

AirEngagePhase6 - processes the intercept phase for the Flexible SAM ruleset. It is scheduled to execute at the intercept time. If the intercept has not been canceled, a determination is made as to the success or failure of the intercept.

CheckIFTUWeapon - evaluates whether to reschedule phase 6, compute new IP, perform kill determination, perform kill assessment, or whether to drop the engagement all together for an IFTU engagement.

KillAssessment - performs kill evaluation for an engagement at intercept time. A message is sent to the commander with the results of the engagement, and the target select phase is scheduled for SAMs in single engagement mode.

InitEngageVolume - is an intercept utility that initializes the engagement volume constraints as a function of target type.

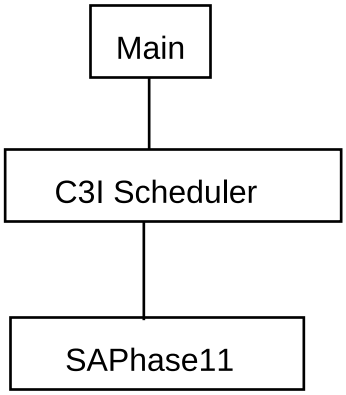

FIGURE 2.11-5. Calling Tree for Flexible SAM Phase 11 (Reload).

Modules Names and Description for Phase 11

SAPhase11- processes the reload phase for the Flexible SAM ruleset. The platform is reloaded with the proper weapon and the number of reloads for this weapon is decremented by one.

Flexible SAM Logic Flow

The Flexible SAM ruleset consists of four phases. Each phase is executed over a period of one or more simulation intervals. In each interval a phase either reschedules itself or transitions to a new phase. Figure 2.11-6 is the transition state diagram for the Flexible SAM. All of the Flexible SAM phases are enabled from within the ruleset itself. Phase 6, the intercept phase can be enabled for a second Flexible SAM platform by an IFTU platform and vice versa.

Phase 1 selects high priority targets to be engaged and then assigns weapons for the engagement. It is scheduled when a track appears in the Flex SAM’s track file and will continue to reschedule itself as long as the track file contains tracks meeting certain criteria (Section 4.7.1 of the EADSIM Methodology Manual). Phase 1 will transition to Phase 17, the counter measures phase if a track is found to be an ARM and the SAM has counter measures capability. Phase 1 will schedule Phase 12, React to SAM Lock, for an AG Attacker platform if the SAM has established lock on the AG Attacker. Phase 1 will transition to Phase 5, SAM Launch, if the SAM is tracking the target and a weapon has been assigned to it.

Phase 5, the launch phase, determines whether a launch can be executed and when it is to occur. Phase 5 will reschedule itself if:

- no launch has yet been scheduled, i.e. a launch record has been established for the target but the launch time has not been determined or
- the engagement uses an IFTU but the IFTU platform has not yet been notified to proceed with the engagement or
- the current target has been assigned to a remote launcher or
- the target is in coast and the time of track validity has not yet expired.

Phase 5 will transition to Phase 6, the intercept phase, if a launch has occurred and the SAM is conducting the intercept. Phase 5 will schedule Phase 6 for an IFTU platform if the IFTU has been assigned responsibility for the intercept. Phase 5 will also schedule Phase 10, Missile Flyout, for an explicit missile when such a missile is launched. Phase 5 will transition to Phase 11, the reload phase, when its weapon count is zero.

In Phase 6, the intercept phase, the engaging platform will perform periodic track continuity checks until the intercept time has been reached. After each check, Phase 6 will reschedule itself for the next track update. If an IFTU platform is conducting the intercept but a Flexible SAM is responsible for the engagement, meaning that it is performing kill assessment, the IFTU platform will schedule Phase 6 for the Flexible SAM.

In Phase11, the reload phase, the SAM's weapon count is incremented by the number of weapons in a reload, the reload count is decremented and control returns to the simulation executive.

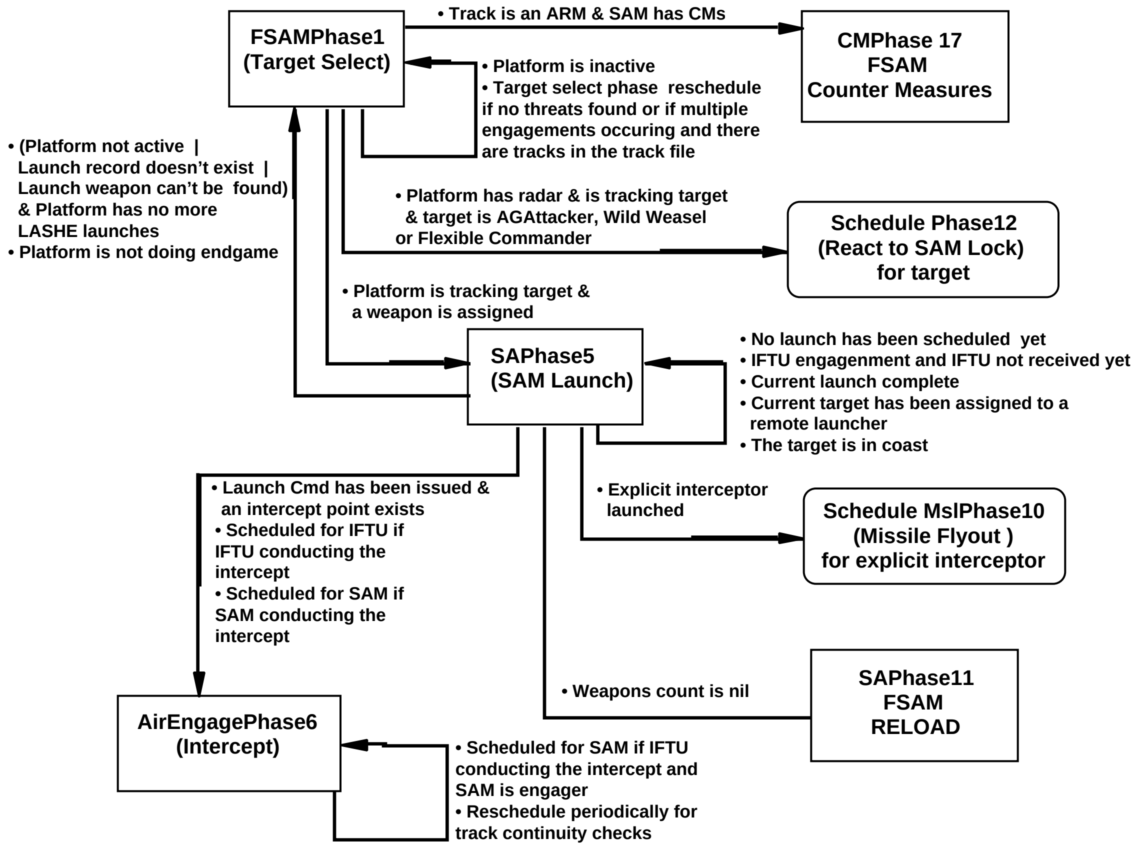
FSAMPhase1
(Target Select)
• Track is an ARM & SAM has CMs
CMPhase 17
FSAM
Counter Measures
• Platform is inactive
• Target select phase reschedule if no threats found or if multiple engagements occuring and there are tracks in the track file
• (Platform not active | Launch record doesn't exist | Launch weapon can't be found) & Platform has no more LASHE launches
• Platform is not doing endgame
• Platform has radar & is tracking target & target is AGAttacker, Wild Weasel or Flexible Commander
Schedule Phase12
(React to SAM Lock)
for target
• Platform is tracking target & a weapon is assigned
SAPhase5
(SAM Launch)
• No launch has been scheduled yet
• IFTU engagenment and IFTU not received yet
• Current launch complete
• Current target has been assigned to a remote launcher
• The target is in coast
• Launch Cmd has been issued & an intercept point exists
• Scheduled for IFTU if IFTU conducting the intercept
• Scheduled for SAM if SAM conducting the intercept
• Explicit interceptor launched
Schedule MslPhase10
(Missile Flyout)
for explicit interceptor
• Weapons count is nil
SAPhase11
FSAM
RELOAD
AirEngagePhase6
(Intercept)
• Scheduled for SAM if IFTU conducting the intercept and SAM is engager
• Reschedule periodically for track continuity checks

FIGURE 2.11-6. Flexible SAM State Diagram.

Logic Diagrams and Module Descriptions

Module Name: FSAMPhase1

The C3I module, FSAMPhase1, is the module that processes the target select phase for the Flexible SAM ruleset. The top-level inputs and outputs for FSAMPhase1 are listed in Table 2.11-3. The only inputs are the current simulation time, SimTime, and a pointer to the engagement order structure. FSAMPhase1 returns a single output variable which contains a completion code. The steps performed by the module are shown in Figure 2.12-7.

TABLE 2.11-3. Inputs and Outputs for FSAMPhase1.

Inputs:	
SimTime	Current Simulation Time
Order	Pointer to Engagement Order Struct
Outputs:	
Return Status	Returns 0 (zero) for success

FIGURE 2.11-7. FSAMPhase1 Logic Flow.

FIGURE 2.11-7. FSAMPhase1 Logic Flow. (Contd.)

FIGURE 2.11-7. FSAMPhase1 Logic Flow. (Contd.)

FIGURE 2.11-7. FSAMPhase1 Logic Flow. (Contd.)

FSAMPhase1 Block Descriptions

Block 1. If platform is dead, exit phase. If platform is inactive, reschedule FSAMPhase1 and exit phase. If track file contains no tracks, exit phase. If platform is line-of-sight (LOS) system with no in-flight target update (IFTU) support, set flags to assess local tracks only.

Block 2. FxAssessThreat is called assess threats to Flexible SAM assets and build an asset-threat category (ATC) list.

Block 3. VerifyLaunchThreat is called to delete pending launches against tracks the platform no longer considers a threat.

Block 4. Loop through TBM tracks in track file and check launch prediction conditions. If the track's launch has been predicted and not sent to commander yet, a message is sent to commander specifying the estimated launch position. If the track flags indicate no launch prediction and the track has been assessed, this indicates that both launch and impact have been predicted. A message is sent to inform the commander that impact position has been predicted for this track.

Block 5. This block refers to the Low Altitude Simultaneous HAWK Engagement (LASHE) mode of operation. LASHE is not addressed.

Block 6. Loop through the asset-threat category (ATC) list that was built by FxAssessThreat.

Block 7 - 9. Get track associated with ATC. If track is a missile, set track flags for TM Threat. If track has been classified as an ARM, schedule ARM Countermeasures.

Block 10. SelectWeapon is called to select best weapon to engage target. SelectWeapon returns an estimated intercept time for the track.

Block 11 - 13. If a weapon was found to engage the target, the fire unit from which the weapon was selected will be scored against fire units which have planned engagements for this target. If committing on a target that is engaged by another platform, the time-to-last-launch threshold is checked to determine if intercept is possible within the threshold. If intercept is possible, then the two engagements are deconflicted.

Block 14. The Flexible SAM Target Select Phase (FSAMPhase1) is rescheduled at it phase start time.

Block 15. Build a launch record for this target/weapon category using the estimated intercept time returned by SelectWeapon and add launch record to the platform launch record list.

Block 16. Determine if an ATC already exists for this track. If an ATC record already exists for this track, clear the ContinueEngagement flag.

Block 17. If ContinueEngagement flag still set, create an ATC record for the subsequent shot for this track, clear the intercept critical threat (ICT) flags to allow ICT recomputation later and store entry in ATC list.

Block 18. SelectWeapon returns an indicator for delayed scheduling of the launch. This is a function of phase repeat time and weapon posture time. If scheduling should be delayed, continue to next track.

Block 19. SchNextLaunch is called to schedule the launch phase for first shot against this track.

Block 20-21. Committing on this target. Set track flags indicating that the track is engaged and I am the engager. Also if this is the first launch for this track, send engagement report (e.g. tell participants on network that I am engaging this track). If the track was command assigned, a WILCO will be sent to the commander.

Block 22. If an engage lead time was returned by the weapon selection logic, EngageControl is called the log a C2_EARLY_ENGAGE action. When possible, engagements are committed early to allow more time for deconfliction to occur.

Block 23. If the track is associated with a commanded launch, set launch/track flags indicating that the commander will be responsible for the engagement and endgame. If track is not a commanded launch, check for IFTU support. If IFTU is required, send a command assignment message to the IFTU platform making the IFTU platform responsible for engagement. If the IFTU platform has been set up for endgame evaluation, set the launch flags indicating IFTU responsible for endgame.

Block 24. If the target is not a missile, the reaction phase for the target is scheduled in response to the SAM lock.

Module Name: FxAssessThreat

The C3I module, FxAssessThreat, is the module used by the Flexible SAM Target Select phase to assess threats to platform assets. The top-level input and output variables and their definitions for FxAssessThreat are listed in Table 2.11-4. The steps performed by the module are shown in Figure 2.11-8.

TABLE 2.11-4. Inputs and Outputs for FxAssessThreat.

Inputs:	
SimTime	Current Simulation Time
OpFac	Pointer to OpFacStruct
ATCList	Pointer to ATC List
ATCListCount	Pointer to ATC List Counter
FlexFlags	Ruleset Flags
Flags	Assess Local/Grunt Flags
Outputs:	
LASHEFLag	LASHE Triggered Status

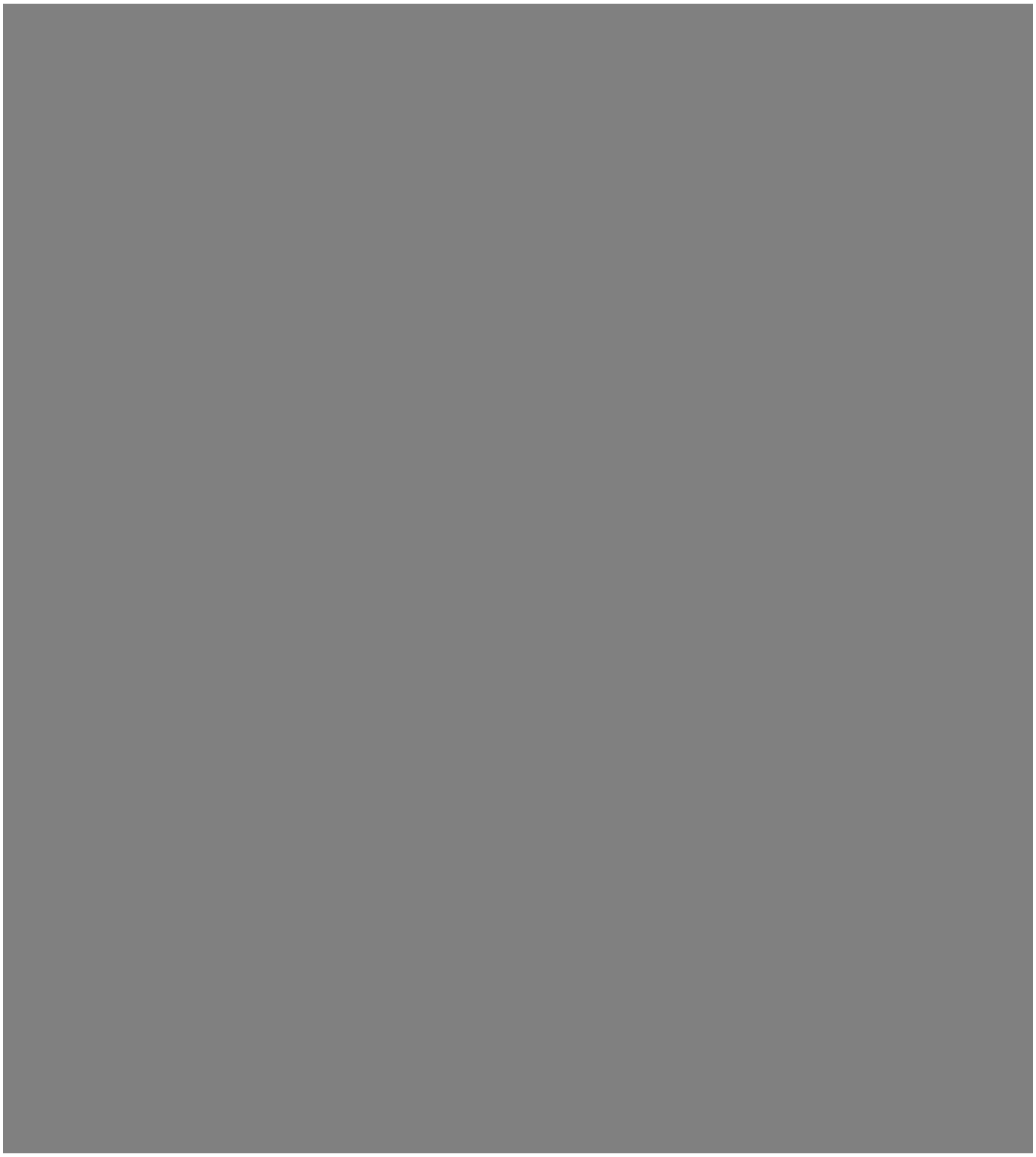

FIGURE 2.11-8. FxAssessThreat Flow Diagram.

FIGURE 2.11-8. FxAssessThreat Flow Diagram. (Contd.)

FIGURE 2.11-8. FxAssessThreat Flow Diagram. (Contd.)

FIGURE 2.11-8. FxAssessThreat Flow Diagram. (Contd.)

FxAssessThreat Block Descriptions

Block 1. This block refers to the Low Altitude Simultaneous HAWK Engagement (LASHE) mode of operation. LASHE is not be discussed in this document.

Block 2. MaxLockTime is used in FxAssessThreat as a test for CANTCO conditions. If a platform locks on a target and does not fire a shot within MaxLockTime, a CANTCO is generated. Set the “multi-shot prioritization” flag and set AOI Flags to indicate the AOI types that will be checked for ABT threats.

Block 3. Tracks that are inactive, ground tracks or delayed tracks are not processed.

Block 4. If the track has been engaged and I am not the engager, continue to next track if it is ABT. If it is TM and engager is in my deconflict group, deconflict the engagement.

Block 5. If launch preemption enabled, clear ShotsFired in ATC record. This will allow for possible preemption of pending launch. If launch preemption is not enabled, set ShotsFired flag to InAirCount (shots in the air). If this track has been locked on for MaxLockTime, send CANTCO to commander.

Block 6. If the track has been assessed as dead, is a HANDOVER track (e.g. tracks are flagged as HANDOVER tracks at the time of a command assignment if the commanded platform does not have this track in its track file), is a friendly track, or is a STALE track, continue to the next track to be processed.

Block 7. If the track is ABT and is on a Low Level Transit Route (LLTR), continue to next track if the current track is a friendly ABT track. If the track was command assigned, send CANTCO to the commander for target on LLTR. If assessing local only mode and track is not a local track, continue to next track.

Block 8. Get the priority for the target associated with this track. If the priority indicates don't shoot (priority = 0), continue to the next track.

Block 9. SetC2Mode is called to determine the operational mode against the target type associated with this track. The operational mode can be centralized, decentralized or autonomous. SetC2Mode also returns the asset types to be defended against ABTs/TMs in the specified operation mode.

Block 10. If track is an ABT and has not been classified as NOT_ARM, a call is made to CheckICTVolumes to determine if target is in an intercept critical target (ICT) volume. If target is in ICT volume and track flags indicate both ARM class and NOT_ARM class, then reclassify track as NOT_ARM. If target is in ICT volume and track flags do not indicate both ARM class and NOT_ARM class and this is the first classification of the track, then classify track as ARM and schedule ARM class countermeasures for this platform.

Block 11. If track is classified as an ARM, loop through all platform assets to determine if any assets are in danger. If the platform ID is same as asset ID, set “self defense” flag in the ATC record. Otherwise, set “asset defense” flag in the ATC record, then continue to next track.

Block 12. If track is a non-ABT track and boost phase intercept (BPI) priority is being applied and final missile burnout time has been reached, continue to the next track. If BPI priority not being applied and missile impact point has not been predicted, FxProcessMissile is called to determine if the impact point can be predicted.

Block 13. If applying BPI priority or the impact point has been predicted, the missile will be processed and an attempt made to select a weapon. If an impact point for this track was determined remotely, FxProcessMissile is called to perform a local computation of the impact point. If the local impact point computation was successful, SelectWeapon is called to select a weapon for engaging this target. Otherwise, continue to the next track to be processed. If no weapon is selected, continue to next track. Prioritize and store this threat in the ATC list. If BPI priority not applied, loop through assets and set "asset/self defense flags". Store Tactical Missile Defense Area (TMDA) threat as a low priority entry in the ATC list if 1) missile impact point is within TMDA, 2) TM zone defense being applied, 3) No assets in danger, and 4) BPI priority not being applied.

Block 14. If BPI priority is being applied and the platform is operating in decentralized mode, call FxProcessMissile to determine if the track assess time has been reached. If track assess time not reached, continue to next track. If the missile is still in boost phase, prioritize and store entry in ATC list. Then continue to the next track.

Block 15. If the track assess time has been reached, set track as ABT_ASSESSED and classify threat as ABT class. If the track assess time has not been reached, continue to next track.

Block 16. If a jammer signal has been detected, loop through the sensors to find the radar that is tracking this target. If the platform is not tracking this target with radar, setup ATC record for Home On Jam (HOJ) launch. Prioritize and store in ATC list.

Block 17. Determine if ABT track is in an appropriate AOI for engagement. If not, continue to next track. For unknown tracks (i.e. not hostile and not friendly), continue to next track if no authority to engage unknowns.

Block 18. If applying BPI priority, SelectWeapon is called to choose a weapon and get an estimated intercept time for the track. If no weapon found for target, continue to next track.

Block 19. If track is command assigned and track is a cruise missile that is not ground engageable, continue to next track. If cruise missile is ground engageable, prioritize and store entry in ATC list for threat and then continue to next track.

Block 20. Loop over assets to determine if any are threatened by this target. If an asset is threatened and ground engageable, prioritize and store entry in ATC list for each asset.

Block 21. If defend mode is ABT zone defense for this target and assets are not in danger, check for unknown track. If unknown track and no authority against unknowns, continue to next track. Otherwise, compute range from track to nearest asset and zone defense entry for track in ATC list.

Module Name: SelectWeapon

The C3I module, SelectWeapon, performs weapon selection for the Flexible SAM ruleset. The top-level inputs and outputs and their definitions for SelectWeapon are listed in Table 2.11-5. The steps performed by the module are shown in Figure 2.11-9.

TABLE 2.11-5. Inputs and Outputs for SelectWeapon.

Inputs:	
OpFac	Pointer to OpFacStruct
Track	Pointer the TrackEntry
SimTime	Current Simulation Time
SalvoFlag	Pointer the TrackEntry
LASHEFlag	Indicates LASHE mode
NoWait	Indicates wait/no wait delay
SetIFTU	Indicates IFTUID should be set
Outputs:	
NShots	Number of shots against target
EngageID	ID of SAM or Remote Launcher
IntTime	Estimated Intercept Time
InterceptPos	Intercept Position
EngageLeadTime	Indication of Early Engagement
PostureDt	Weapon Posture Time
IllumTime	Illumination Time Req. by Weapon
IFUTID	IFTU Platform ID

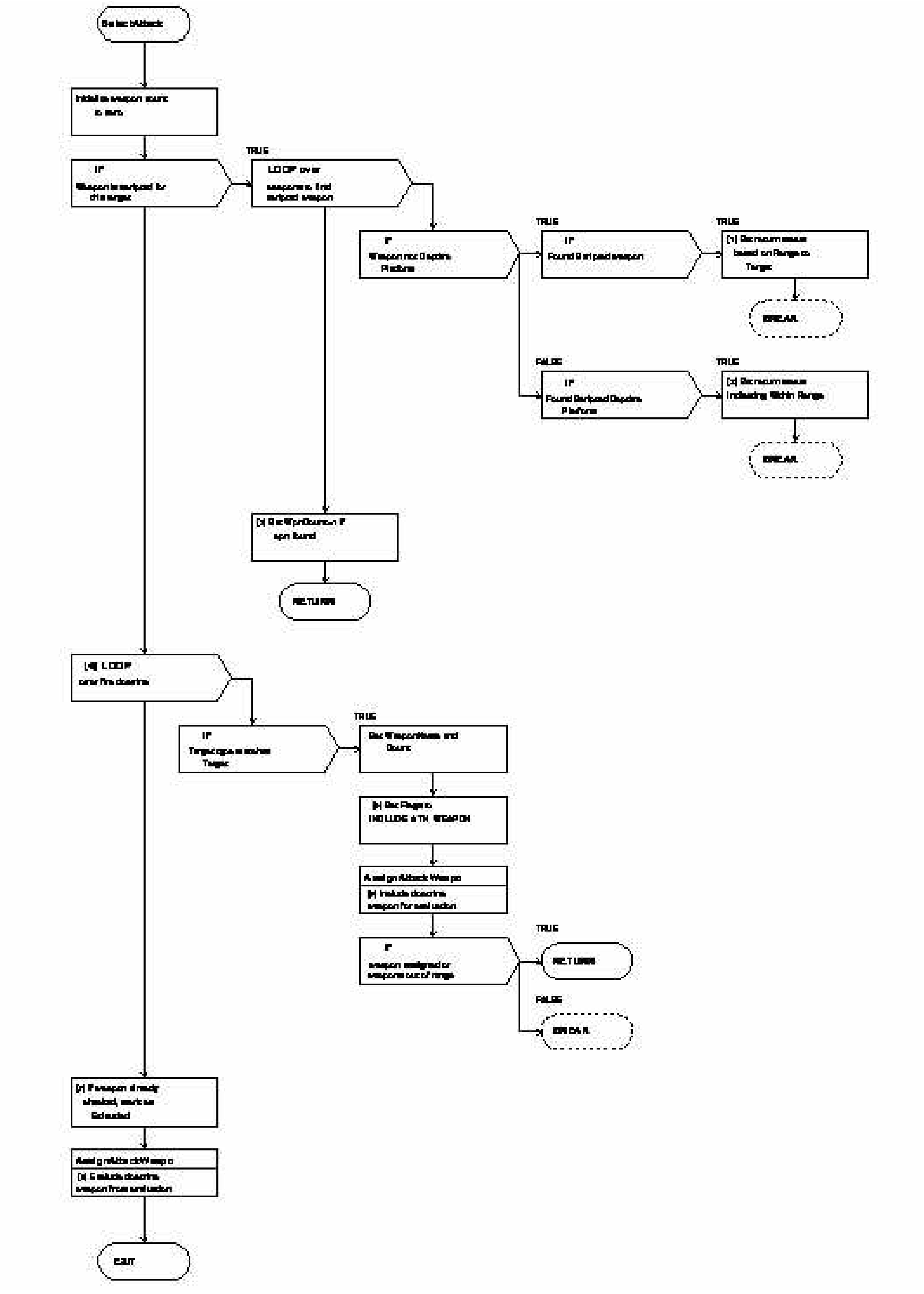
SelectAttack
Initialize weapon count to zero
IF
Weapon is scripted for this target
TRUE
LOOP over
weapons to find scripted weapon
IF
Weapon not Depletes Platform
TRUE
IF
Found Scripted weapon
TRUE
[1] Set return value based on Range to Target
BREAK
FALSE
IF
Found Scripted Depletes Platform
TRUE
[2] Set return value Indicating Within Range
BREAK
[3] Set WpnCount to 1 if wpn found
RETURN
[4] LOOP
over fire doctrine
IF
Target type matches Target
TRUE
Set WeaponName and Count
[5] Set Flag to INCLUDE 6TH WEAPON
Assign Attack Weapon
[6] Include doctrine weapon for evaluation
IF
weapon assigned or weapons out of range
TRUE
RETURN
FALSE
BREAK
[7] If weapon already checked, mark as Excluded
AssignAttackWeapon
[8] Exclude doctrine weapon from evaluation
EXIT

FIGURE 2.11-9. SelectWeapon Flow Diagram.

Block Descriptions

Block 1. If track is a TM track and time to first launch is less than two phase repeat times, exit SelectWeapon. If track is ABT track and SAM engagement restrictions are enabled, check receding/approaching for CANTCO condition. If ABT track is receding and user has restricted receding, set track "missed engage" flag for CANTCO and exit SelectWeapon. If ABT track is approaching and user has restricted approaching track "missed engage" flag for CANTCO and exit SelectWeapon.

Block 2. Determine if local platform radar is tracking target, this information will be used later during weapon evaluation.

Block 3. If SalvoFlag is set the following rules will be used for weapon selection: The first weapon choice is the same weapon ID as previously launched. If this weapon ID not available, second choice is a weapon on the SAM or remote launcher that has the same weapon element definition as the previously launched weapon. If there is more than one of these, the shortest intercept time will determine weapon selection. If neither the same weapon nor a weapon with same weapon element definition is available, standard weapon selection logic is applied excluding those weapon already checked in previous tests.

Block 4. If no weapon has been selected to this point, loop through weapons.

Block 5. If HOJ launch specified for this track, loop over IFTU phases for weapon to find active guidance mode, set "missile can activate seeker" flag. If no weapon with active seeker mode is found, continue to next weapon.

Block 6. Evaluate weapon to determine intercept time, illumination time required, and time to first launch.

Block 7. If weapon type found was laser, set number of shots against target to one. Otherwise, check LASHE target/weapon conditions.

Block 8. If track is a TM track and no weapon is found to intercept the target, check for possible future intercept. If no future intercept possible, count pending launches. If launch count equals in-air count and launching platform is not a laser, set the track as unengageable. If future intercept is possible, set "time to first launch" for track.

Block 9. If engagement requires launch preemption, remove the preempted launch from launch list for platform.

Block 10. Store the intercept position, time, and engage lead time for return to calling routine.

Module Name: SAPhase5

The C3I module, SAPhase5, processes the launch phase for the Flexible SAM ruleset. The top-level inputs and outputs and their definitions for SAPhase5 are listed in Table 2.11-6, and the detailed flow diagram is illustrated in Figure 2.11-10.

TABLE 2.11-6. Inputs and Outputs for SAPhase5.

Inputs:	
SimTime	Current Simulation Time
Order	Pointer to Engagement Order Struct
Outputs:	
Return Status	Returns 0 (zero) for success

FIGURE 2.11-10. SAPhase5 Flow Diagram.

FIGURE 2.11-10. SAPhase5 Flow Diagram. (Contd.)

FIGURE 2.11-10. SAPhase5 Flow Diagram. (Contd.)

Block Descriptions

Block 1. If platform not active, stop launch for NOTACTIVE platform. EngageControl is called to log the C2_STOP_NOTACTIVE action. The launch record is deleted. If an IFTU platform is responsible for the engagement, a CMD_COMPLETE message is sent over the simulated communication network to the IFTU platform for the target associated with this launch. The phase is exited.

Block 2. If the launch record does not exist for the LaunchID specified in the Order structure, a CMD_COMPLETE message is sent to the IFTU platform for the target associated with this. If platform is in LASHE mode and no longer has any LASHE launches, the target select phase (FSAMPhase1) is rescheduled at its phase start time. SchNextLaunch is called to reschedule the launch phase.

<u>Block 3.</u> If the weapon associated with the launch record does not exist or weapon count is zero, stop the launch due to NOWEAPONS. EngageControl is called to log a C2_NOWEAPONS action, a CMD_COMPLETE message is sent to the IFTU platform responsible for the engagement, the launch record is deleted and the phase is exited.

<u>Block 4.</u> If a Flexible SAM is performing the launch, set launcher to "Flexible SAM". Otherwise, set launcher to "remote launcher" and check the remote launcher for active status. If remote launcher is not active, EngageControl is called to log a C2_STOP_RL_NOTACTIVE action.

<u>Block 5-6.</u> Get the track entry pointer from the track file for this target ID. Clear the command variable. The command variable will be set subsequently indicating one of many conditions tested (abort launch, continue launch, stop no track, etc.)

<u>Block 7.</u> If an IFTU platform is responsible for the engagement, check IFTU response to the previous command assignment for this engagement. If the response was a CANTCO, set command to C2_STOP_NO_SUPPORT. If neither WILCO nor CANTCO response was received, check to see if the IFTU support response time has expired. If response time expired, set command to C2_STOP_NO_SUPPORT. If time has not expired, call SchNextLaunch to reschedule launch phase, then exit phase. If command is set to C2_STOP_NO_SUPPORT, EngageControl is called to log the stop launch action and StopSAMLaunch is called to stop the launch for no support.

<u>Block 8.</u> If the command variable is not set, CheckCoast is called the check if the assigned launch wait delay has been exceeded. If launch wait delay has been exceeded, CheckCoast returns a value for the command variable of C2_STOP.

<u>Block 9.</u> If track entry was not found, set command to C2_STOP_NO_TRACK.

<u>Block 10.</u> If command is set to C2_LAUNCH, this indicates that we are continuing with the launch. The planned resources to support this launch are removed from the list of available resources.

<u>Block 11.</u> If the target is a TM, compute intercept time for tactical missile. If target is ABT (cruise missiles are considered ABTs), compute intercept for the ABT.

<u>Block 12.</u> If an intercept exists for this target, store the intercept time, position and resources to support the launch.

<u>Block 13.</u> If Flexible SAM is launching the weapon, EngageControl is called to log the C2_LAUNCH action.

<u>Block 14.</u> If IFTU platform is responsible for engagement, schedule intercept phase for IFTU platform. If IFTU platform is setup for endgame evaluation responsibility, set IFTU_ENDGAME launch flag.

<u>Block 15.</u> If interceptor is explicitly flown, schedule the flyout phase for the explicit missile interceptor. If an explicit network is specified for interceptor, UpdateNets is called to create a dynamic network for communication between the explicit missile interceptor and its commander.

Block 16. The intercept phase is scheduled for this platform, the weapon count is decremented and SchNextLaunch is called to reschedule the launch phase for this platform.

Block 17. If Flexible SAM not performing launch, a command assign message is sent to the remote launcher to launch against this target. SchNextLaunch is called to reschedule the launch phase for the Flexible SAM.

Block 18. If command is C2_STOP or C2_ABORT, EngageControl is called to log the stop action and StopSAMLaunch is called to stop the launch.

Block 19. If command is C2_COAST, EngageControl is called to log the COAST action and SchNextLaunch is called to reschedule the launch phase for this platform.

Module Name: TBMIntercept

The C3I module, TBMIntercept, is used by SAPhase5 to estimate intercept time for a TBM engagement. The top-level inputs and outputs and their definitions for IBMIntercept are listed in Table 2.11-7 and the steps performed by the module are shown in Figure 2.11-11.

TABLE 2.11-7. Inputs and Outputs for TBMIntercept.

Inputs:	
ID	ID of Platform computing Intercept
SensorOpFacID	Owner of Sensor
PostureTime	Weapon Posture Time
MissileID	ID of the TBM being intercepted
SimTime	Current Simulation Time
Weapon	Pointer to weapon structure
Flags	Constraint Flags
SetIFTU	Indicates setting of IFTUID
Outputs:	
LaunchTime	Seconds until launch
IntTime	Seconds until intercept
IntRange	Intercept Range
InterceptPos	ECI Position of Intercept
TempList	Resource List
IFTUID	IFTU Platform ID
WeaponPK	Weapon Probability of Kill
IllumTime	Illumination time required by weapon

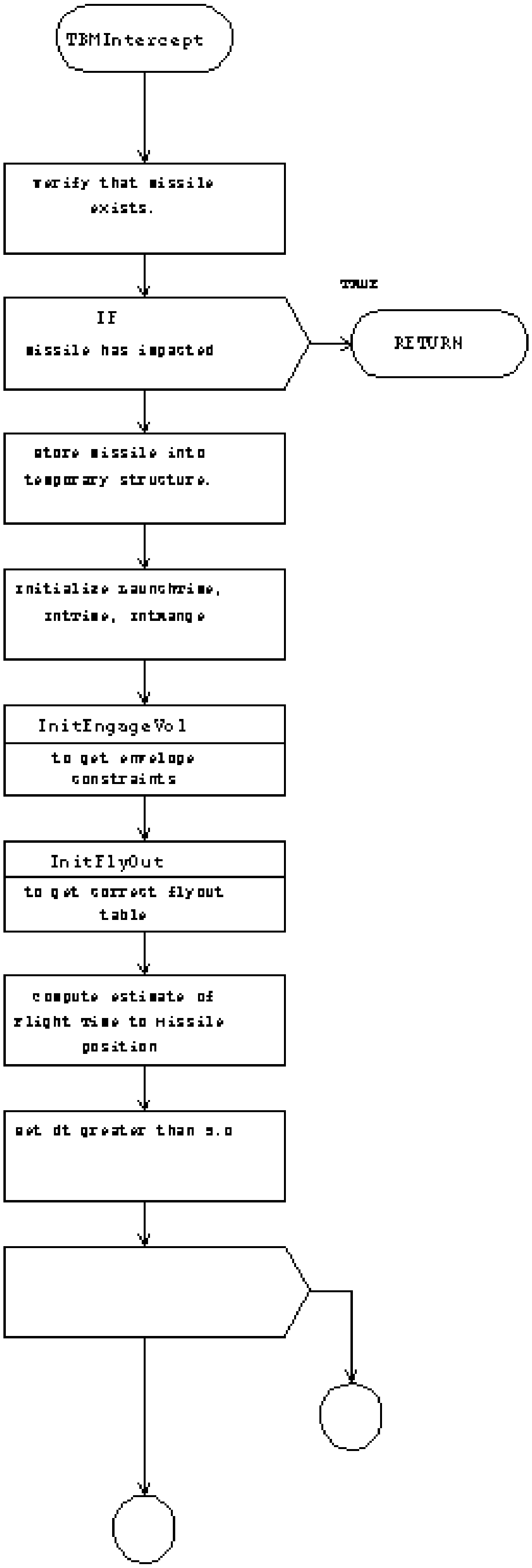
TBMIntercept
verify that missile exists.
TRUE
IF
missile has impacted
RETURN
store missile into temporary structure.
initialize LaunchTime, IntTime, IntRange
InitEngageVol
to get envelope constraints
InitFlyOut
to get correct flyout table
compute estimate of flight time to missile position
set dt greater than 5.0

FIGURE 2.11-11. TBMIntercept Flow Diagram.

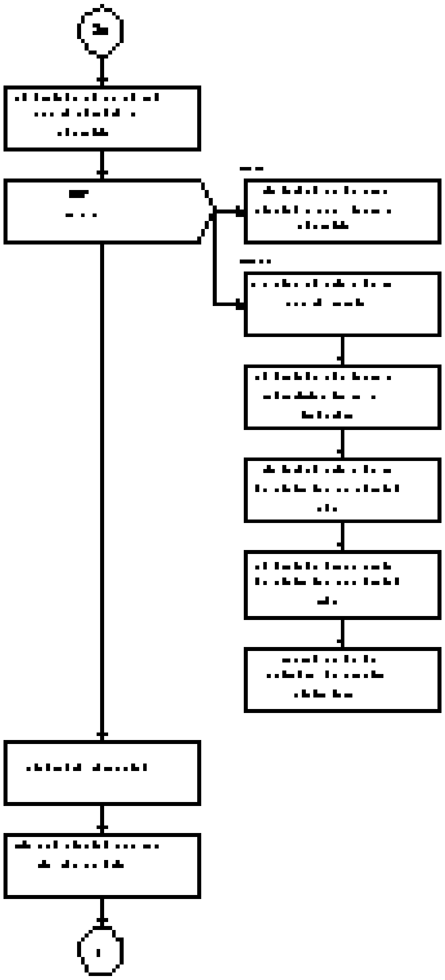

FIGURE 2.11-11. TBMIntercept Flow Diagram. (Contd.)

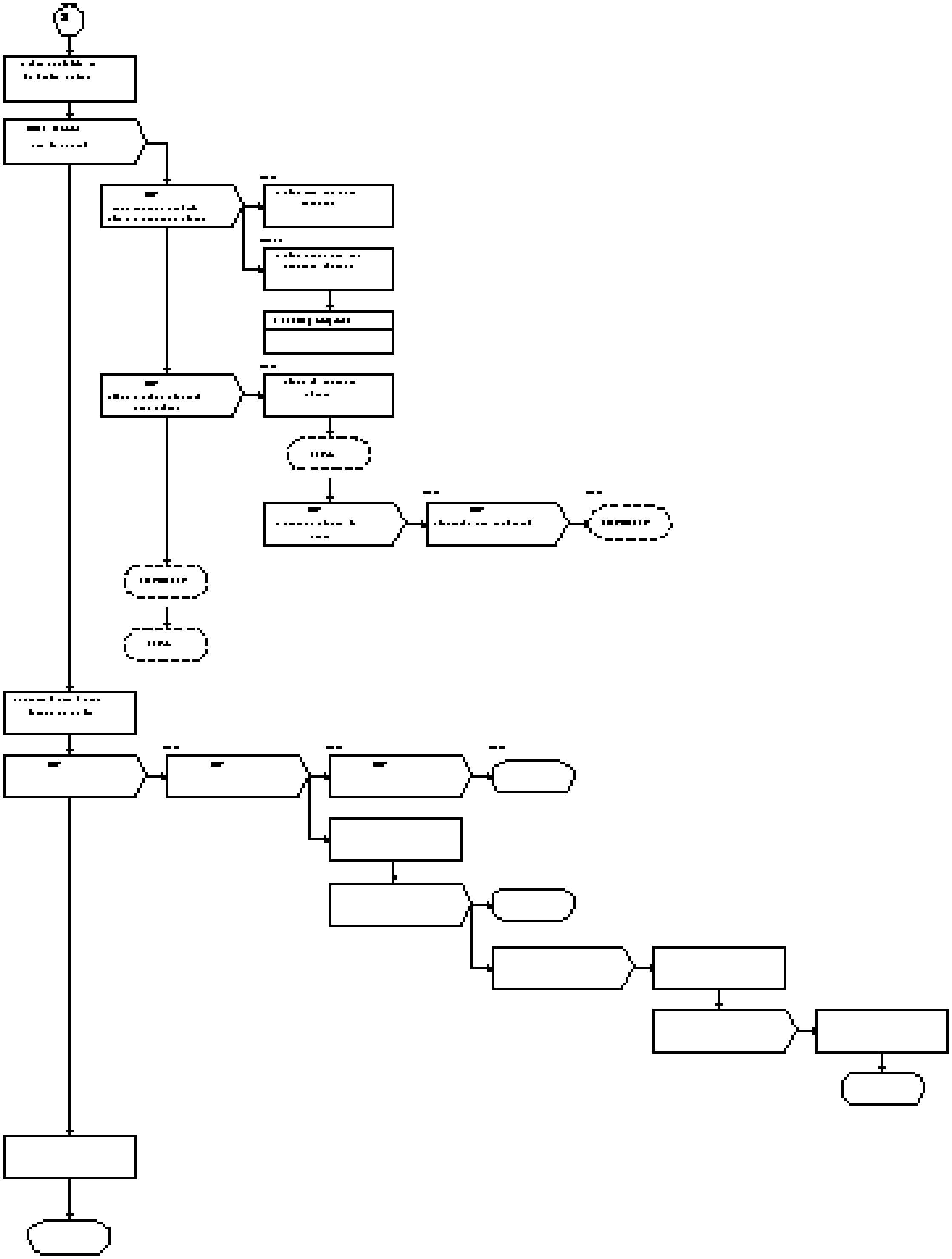

FIGURE 2.11-11. TBMIntercept Flow Diagram. (Contd.)

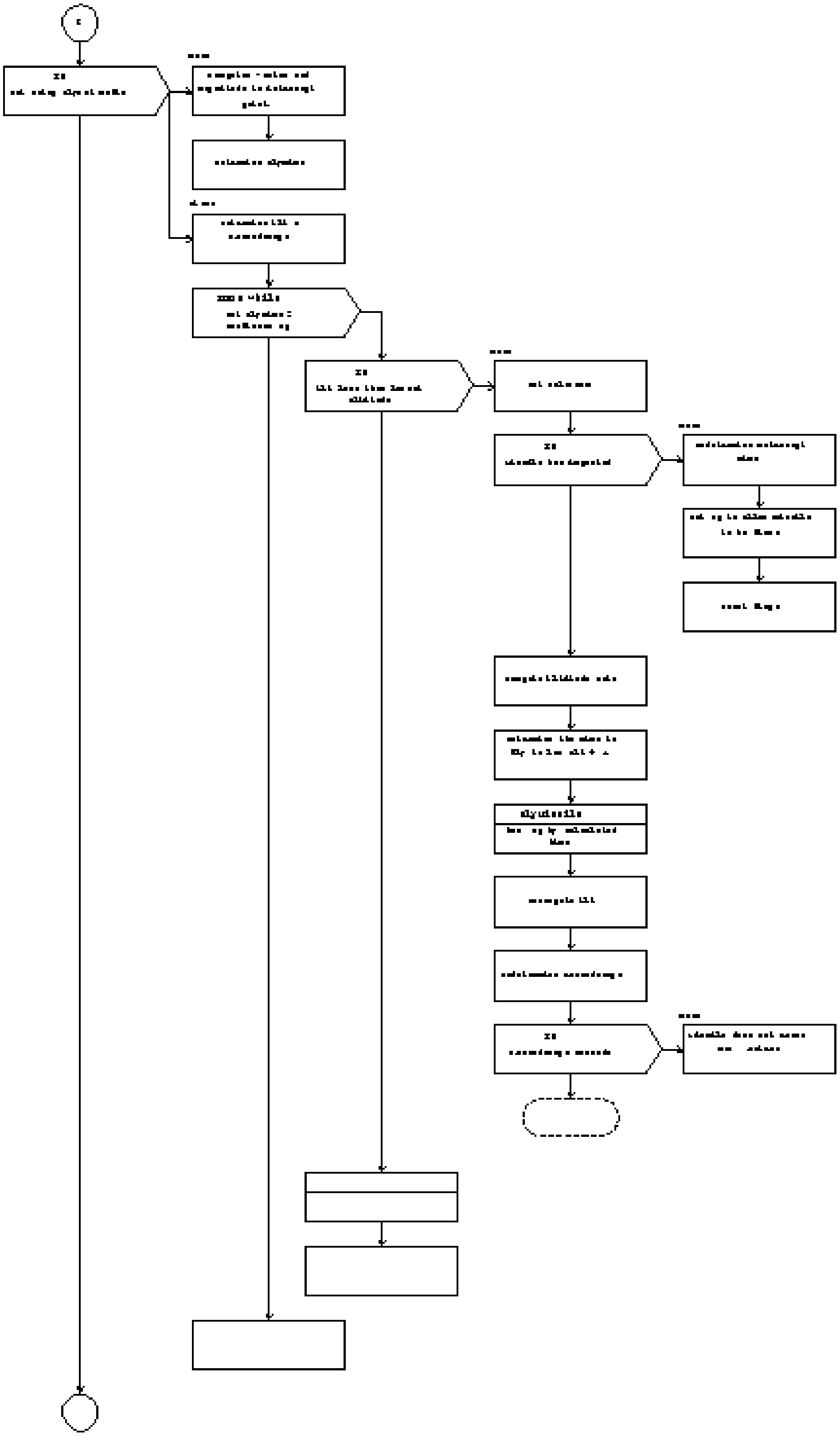

FIGURE 2.11-11. TBMIntercept Flow Diagram. (Contd.)

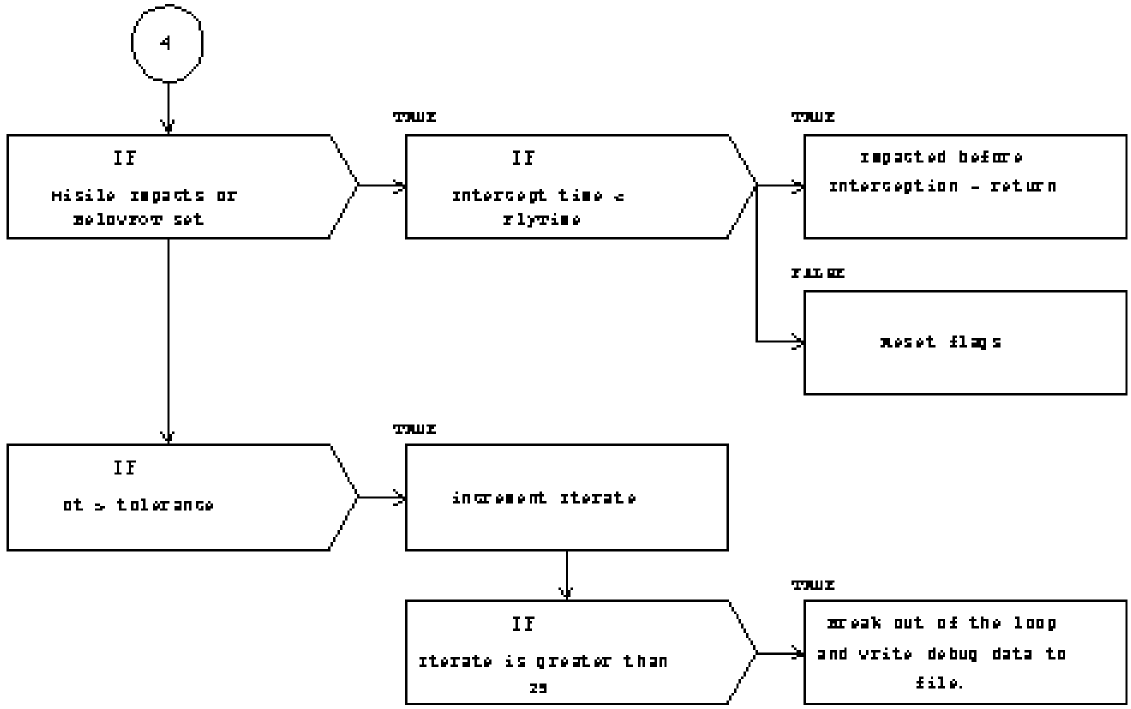
4
IF
Misile impacts or belowTOT set
TRUE
IF
intercept time < flyTime
TRUE
impacted before interception - return
FALSE
reset flags
IF
dt > tolerance
TRUE
increment iterate
IF
iterate is greater than 25
TRUE
break out of the loop and write debug data to file.

FIGURE 2.11-11. TBMIntercept Flow Diagram. (Contd.)

Block Descriptions

Block 1. To establish missile flight iteration parameters, InitEngageVol is called to get engagement volume constraints and InitFlyOut is called to get the weapon flyout table pointer for this missile/target type. An estimate of the time to fly the missile to it's current position is computed.

Block 2. Loop while still iterating on an intercept solution using the weapon flyout table for this missile/target type.

Block 3. Compute deltatime to fly missile forward for this iteration as a function of the ratio of the weapon to missile velocity.

Block 4. Fly the missile forward by delta time.

Block 5. If only the intercept time is needed by the calling routine, exit this module returning the estimated intercept time.

Block 6. Continue flying the missile in one second intervals while the missile is not in a user specified engagement volume. If engagement constraints are violated or weapon probability of kill (PK) is less than engagement volume PK, continue flying the missile. If sensor resources are required and are not available, continue flying the missile. If the weapon requires illumination support and an illuminator is not available, continue flying missile. This loop is exited if the missile enters an engagement volume or if it is determined that the missile will not enter an engagement volume (i.e. no future checks required).

Block 7. The range from the weapon to the target is computed after the target enters an engagement volume.

Block 8. Pre/Post boost phase launch criteria are checked. If the missile is in the boost phase and the weapon requires post boost phase intercept, exit the module returning no intercept solution. If the missile is in the post boost phase and the weapon requires boost phase intercept, exit the module returning no intercept. If the missile is in the post apogee phase and the weapon requires a pre apogee phase intercept, exit this module returning no intercept solution.

Block 9. Set the return values for estimated intercept position and time.

Block 10. Exit the module returning values indicating that the missile is either inside or outside an engagement volume.

Block 11-12. If a flyout table is not being used for the weapon, compute the range from the weapon to the missile and flytime using range, weapon velocity and weapon posture time.

Block 13. Compute the missile altitude and ground range from the weapon to the missile to begin flying missile toward flyout table.

Block 14. Continue flying missile in three second intervals until it is on the weapon flyout table.

Block 15. If the target missile is at an altitude below the minimum flyout table altitude, compute the flytime to fly the missile back to the minimum flyout table altitude. Fly the missile back by the computed flytime. After flying the missile back, determine if the missile position is on the flyout table. If not, exit the module returning no intercept time. The missile does not cross the flyout table.

Block 16. Fly the missile forward for three seconds.

Block 17. If the missile has impacted the ground, exit this module return no intercept. If the maximum weapon flyout time is exceeded, exit the loop.

Block 18. If the maximum weapon flyout time is exceeded, exit the loop.

Block 19. Advance the flytime by the weapon posture time.

Block 20. If weapon impacts before intercept, exit this module returning no intercept.

Module Name: InterceptTime3

The C3I module, InterceptTime3, prepares the “react to lock” phase for the AGAttacker ruleset. The top-level inputs and outputs and their definitions for InterceptTime3 are listed in Table 2.11-8 and the steps performed by the module are shown in Figure 2.11-12.

TABLE 2.11-8. Inputs and Outputs for InterceptTime3.

Inputs:	
OpFac	Pointer to OpFac Structure
WeaponOpFac	Pointer to WeaponOpFac Structure
Weapon	Pointer to weapon structure
PostureTime	Weapon Posture Time
Track	Pointer to TrackEntry
LaunchVel	Launching platforms velocity
WaitDelay	Early engagement delay time
SimTime	Current Simulation Time
Flags	Constraint Checks
SetIFTU	Indicates setting of IFTUID
Outputs:	
Time	Seconds until intercept
InterceptPos	ECI Position of Intercept
LaunchTime	Weapon Launch Time
TempList	Resource List
IFTUID	IFTU Platform ID
WeaponPK	Weapon Probability of Kill
IllumTime	Illumination time required by weapon

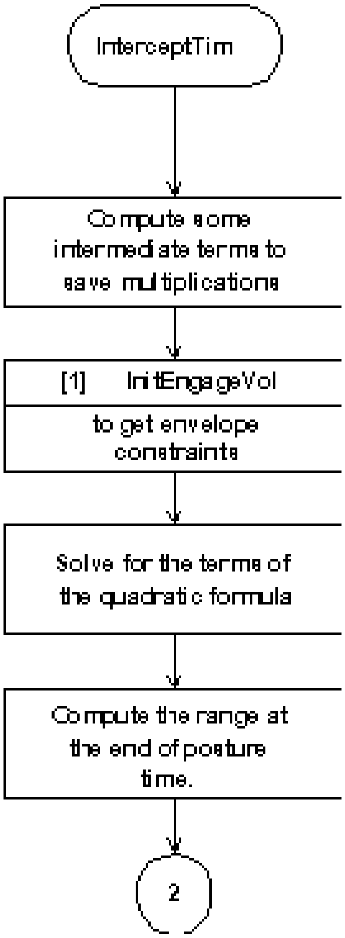
InterceptTim
Compute some intermediate terms to save multiplications
[1] InitEngageVol
to get envelope constraints
Solve for the terms of the quadratic formula
Compute the range at the end of posture time.
2

FIGURE 2.11-12. InterceptTime3.

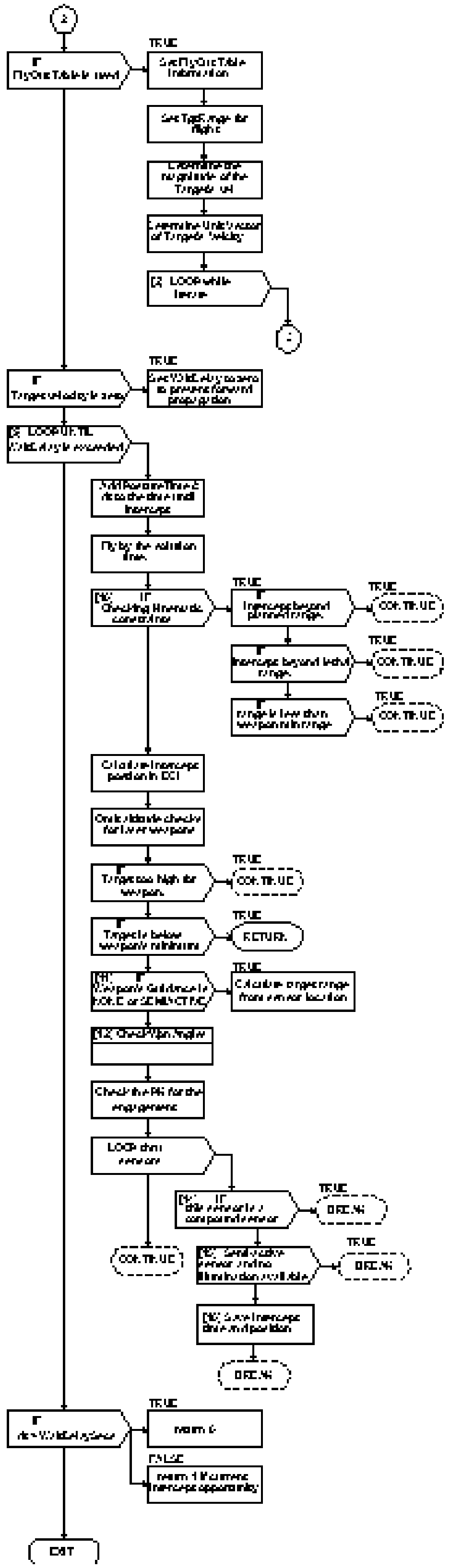
2
TRUE
IF
FlyOutTable is used
Get FlyOutTable information
Get TgtRange for flight
Determine the magnitude of the TargetVel
Determine Unit Vector of Target Velocity
[3] LOOP while iterate
3
TRUE
IF
Target velocity is zero
Set VelDelay to zero to prevent forward propagation
[5] LOOP UNTIL
VelDelay is exceeded
Add ReactionTime & dt to the time until intercept
Fly by the solution time
[10] IF
Checking kinematic constraints
TRUE
IF
Intercept beyond planned range
TRUE
CONTINUE
IF
Intercept beyond lethal range
TRUE
CONTINUE
IF
range is less than weapon min range
TRUE
CONTINUE
Calculate intercept position in ECI
Only validate checks for laser weapons
IF
Target too high for weapon
TRUE
CONTINUE
IF
Target is below weapon's minimum
TRUE
RETURN
[11] IF
Weapon's Guidance is NONE or SEMIACTIVE
TRUE
Calculate target range from sensor location
[12] Check Pan Angles
Check the Pk for the engagement
LOOP thru sensors
[13] IF
this sensor is a companion sensor
TRUE
BREAK
[14] Semiactive sensor and no illumination available
TRUE
BREAK
[15] Save intercept time and position
BREAK
CONTINUE
IF
dt > VelDelayTime
TRUE
return 0
FALSE
return 1 if current intercept opportunity
EXIT

FIGURE 2.11-12. InterceptTime3. (Contd.)

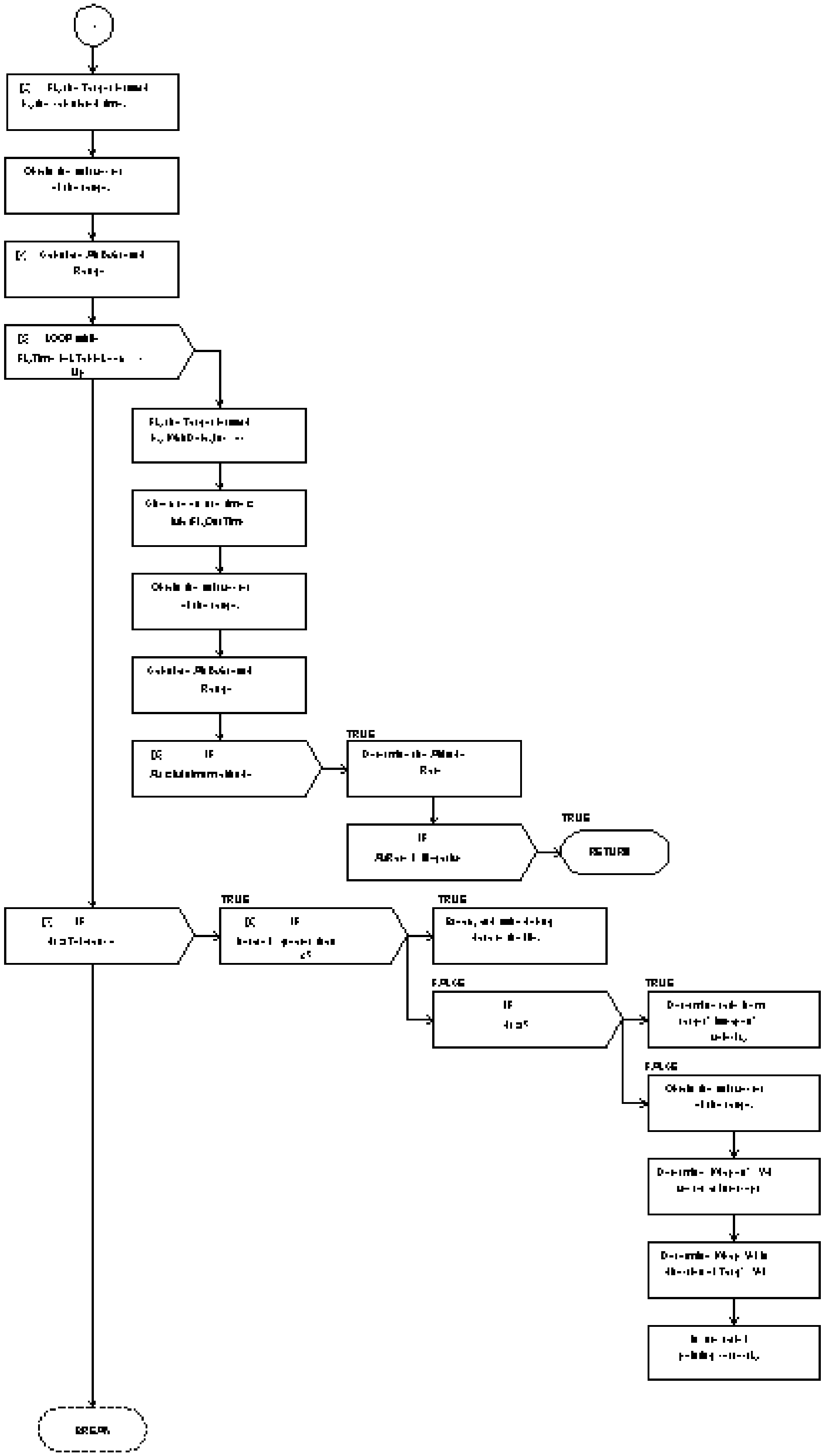

FIGURE 2.11-12. InterceptTime3. (Contd.)

Block Descriptions:

Block 1. InitEngageVol is called to establish engagement volume constraints.

Block 2. Continue looping until the target has reached the region covered in the flyout table.

Block 3. Fly the target toward the flyout table by deltatime. DeltaTime is initially the rough time to intercept obtained by solving the intercept algorithm.

Block 4. Compute the target altitude and range from weapon position for flyout table lookup.

Block 5. Continue looping while the target is not on the flyout table. The target is flown forward in one second intervals. At each interval, the max flyout time is checked and the target altitude and range from weapon position are computed.

Block 6. If the target is below the minimum flyout table altitude, a determination is made as to whether the target will cross the flyout table at all. If not, this module is exited returning no intercept.

Block 7-8. dt represents the difference between the time for the interceptor to reach the intercept point and the time for the target to reach the intercept. If dt greater than Tolerance (e.g. 0.3 seconds), a check is made for max iterations and deltatime is recomputed for the next iteration. Otherwise, the solution is within the tolerance and exit the loop.

Block 9. A viable intercept solution has been found. This loop is used to determine if the weapon can in fact achieve the intercept. The weapon will be flown forward in one second intervals.

Block 10. If the weapon is a laser weapon and the weapon's maximum intercept range has not been exceeded, continue flying target forward. If the weapon is not a laser, the kinematic constraints are checked (i.e. weapon lethal range, min/max engagement volume range). If constraint checks fail, continue flying target forward.

Block 11. The weapon azimuth and elevation constraints are checked. If angle checks fail, continue flying target forward.

Block 12. The intercept constraints specified for the weapon are checked. These constraints include, minimum crossing angle, maximum crossing angle, closing velocity, and weapon look angle. If any of these constraints fail, continue flying the target forward.

Block 13. The weapon PK is computed. If the computed PK is less than the threshold PK specified in the engagement volume, continue flying the target forward.

Block 14. If a sensor resource is required to support the intercept and the resource is not available, continue flying the target forward.

Block 15. If illumination is required to support the intercept and illumination is not available, continue flying the target forward.

Block 16. Return values for intercept time, position and launch time.

Module Name: AirEngagePhase6

The C3I module, AirEngagePhase6, processes the intercept phase for the Flexible SAM ruleset. The top-level inputs and outputs and their definitions for AirEngagePhase6 are listed in Table 2.11-9 and the steps performed by the module are shown in Figure 2.11-13.

TABLE 2.11-9. Inputs and Outputs for AirEngagePhase6.

Inputs:	
SimTime	Current Simulation Time
Order	Pointer to Engagement Order Struct
Outputs:	
Return Status	Returns 0 (zero) for success

FIGURE 2.11-13. AirEngagePhase6.

FIGURE 2.11-13. AirEngagePhase6. (Contd.)

FIGURE 2.11-13. AirEngagePhase6. (Contd.)

FIGURE 2.11-13. AirEngagePhase6. (Contd.)

Block Descriptions:

Block 1. InitEngageVol is called to establish engagement volume constraints.

Block 2. If IFTU phases are defined for the weapon, CheckIFTUWeapon is called to evaluate the intercept.

Block 3. CheckIFTUWeapon determines whether to reschedule the intercept phase, compute a new intercept point, perform a kill determination, perform a kill assessment, or stop the engagement altogether. If CheckIFTUWeapon returns a status “exit intercept phase”, exit this module without performing kill determination and kill assessment.

Block 4. If the weapon guidance mode is semi-active or NLOS, an “intercept failed” message is written to the log file. If a launcher is required for the launch, EngageControl is called to log a C2_FAIL_NO_SUPPORT action for the launcher and a C2_NORMAL action for the Flexible SAM. If a launcher is not required, EngageControl is called to log a C2_FAIL_NO_SUPPORT action for the Flexible SAM. The launch record is deleted.

Block 5. If the engaging platform is evaluating this intercept and the engager is not active, log an intercept failure as follows; If the weapon guidance mode is semi-active or NLOS, an intercept failed message is written to the log file. If a remote launcher is required by the engaging platform for the launch, EngageControl is called to log a C2_FAIL_NOTACTIVE action for the launcher platform and a C2_NORMAL action for the engager platform. If a launcher is not required, EngageControl is called to log a C2_FAIL_NOTACTIVE action for the engager platform. The launch record is deleted. A CMD_COMPLETE message is sent over the simulated network to the IFTU platform due to failed intercept.

Block 6. If the target is dead, set Command to C2_FAIL_DIED. If the target track is in the track file and track is friendly, set Command to C2_ABORT_FRIENDLY. If the track if track is too old, set Command to C2_FAIL_NOTRACK. If the track is not a local track, set Command to C2_FAIL_NO_LOCAL_TRACK.

Block 7. A check is made for performing a kill determination. A kill determination is not performed if any of the conditions in Block 6. are met.

Block 8. Compute the altitude and range from the weapon launch position to the TM target. Get weapon lethal range.

Block 9. Compute the altitude and range from the weapon launch position to the ABT target. Get weapon lethal range.

Block 10. Compute weapons point of closest approach to the target.

Block 11. If the range from the weapon to the target is greater than the weapon's lethal radius, set Command to C2_FAIL_RANGE.

Block 12. If the target is within the lethal range of the weapon, determine the probability of kill. Otherwise, set Command to C2_FAIL_RANGE.

Block 13. Perform a random draw to determine the probability of kill for this target. If kill is successful, set Command to C2_SUCCESS. Otherwise, set Command to C2_FAIL_PK (e.g. failure due to probability of kill).

Block 14. If the target has not been intercepted and is not already dead, set Command to C2_FAIL_NOTRACK.

Block 15. The target should have been intercepted at this point. If not, failure messages are logged. If a remote launcher is launching against the target, EngageControl is called to log the intercept failure Command for the launcher and to log a C2_NORMAL for the Flexible SAM. If Flexible SAM is launching against the target, EngageControl is called to log the failure Command for the SAM. The launch record is then deleted.

Block 16. If the weapon platform is not evaluating this intercept, EngageControl is called to log a C2_STOP action for the Flexible SAM.

Block 17. If an IFTU platform is not responsible for endgame evaluation, schedule the intercept phase for the weapon platform immediately and exit this module.

Block 18. If the target is a TM, AssessTMEngage is called to perform the kill assessment for the TM engagement. This assessment is based on random draws and probabilities of false alarm.

Block 19. If an IFTU platform is responsible for the engagement, send a CMD_COMPLETE message over the simulated network.

Block 20. If the platform evaluating this intercept is the weapon platform, check the gun re-engage conditions. If the platform is active and the weapon is a gun and if the target was not assessed success, dead or destroyed, build new launch records and set the Order engagement structure flags to indicate re-engage with gun weapon.

Block 21. The kill assessment delay is computed and the kill assessment model is scheduled to occur at the current simulation time + kill assessment delay.

Module Name: SAPhase11

The C3I module, SAPhase11, is the module that processes the reload phase for the Flexible SAM ruleset. The top-level inputs and outputs and their definitions for SAPhase11 are listed in Table 2.11-10 and the steps performed by the module are shown in Figure 2.11-14.

TABLE 2.11-10. Inputs and Outputs for SAPhase11.

Inputs:	
SimTime	Current Simulation Time
Order	Pointer to Engagement Order Struct
Outputs:	
Return Status	Returns 0 (zero) for success

FIGURE 2.11-14. SAPhase11 Logic Flow.

Block Descriptions

Block 1. If the platform that is being reloaded is inactive, exit the phase.

Block 2. Loop through the platform weapons to find the weapon ID to be reloaded.

Block 3. If the weapon ID is found, reset the weapon count for this weapon to the user specified count per reload. EngageControl is called to log the C2_RELOAD action. The number of reloads for this weapon ID is decremented and the platform OUT_OF_WEAPONS flag is cleared. If no matching weapon ID was found, log a no weapon found error message and exit phase.

2.11.3.2 AGATTACKER Ruleset

Calling Tree

The calling tree for the Air to Ground Attacker consists of multiple trees corresponding to each of the 8 phases of the ruleset. The calling trees are shown in Figures 2.11-15 through 2.11-22. Each tree begins with the main program for the phase which is invoked by the C3I process scheduler. Each phase is scheduled by another phase, either of the AG Attacker of by another platform. The details of phase transition are discussed in section 2.1.3.4.1. The unique functions that appear in each calling tree are discussed with figure. Some, like CheckAOIPatter, appear in more than one tree but provide the same information or perform the same service and are discussed on their first appearance. EngageControl is a utility function that sends different information to the Flight Processing process from each phase and discussed each time it appears in a calling tree.

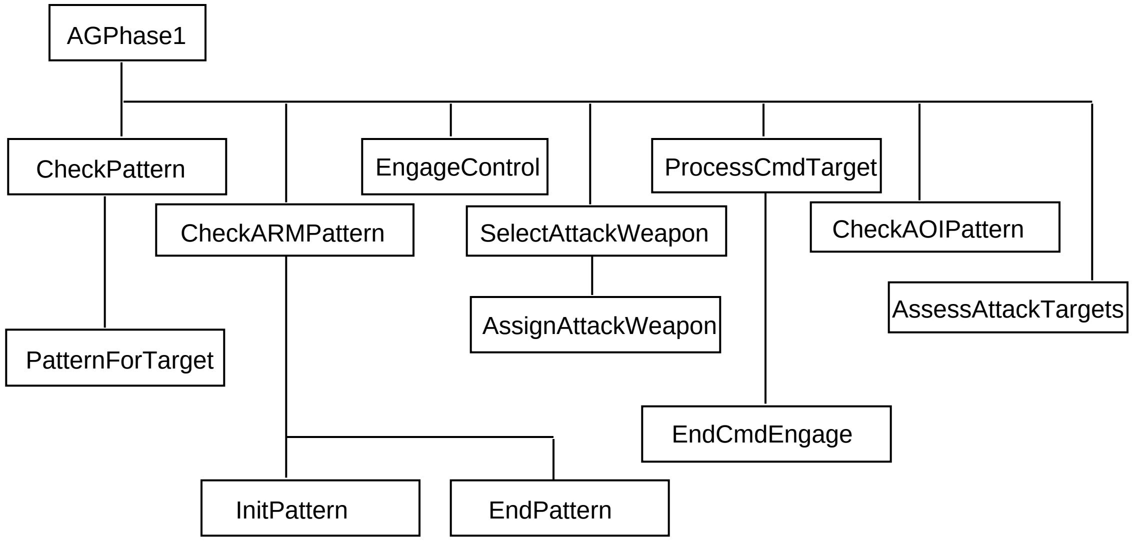

FIGURE 2.11-15. AG Attacker Phase 1 Calling Tree: Target Select.

Module Names & Descriptions for Phase 1

AGPhase1- AGPhase1 is the module that processes the target select phase for the agattacker ruleset. The target select phase selects the next target an AGAttacker flight will engage. AGAttacker flight leaders execute the target select phase to evaluate scripted, detected, and commanded targets. Wingmen only execute the target select phase if they have scripted targets or receive a commanded assignment from their flight leader.

CheckPattern- CheckPattern is the module used by AGAttacker to determine if a search/track profile should be initiated against a ground target. CheckPattern determines if a search profile should be initiated or if the platform should transition from a search profile to a track profile.

PatternForTarget-Determines if a pattern should be initiated for the current target based on user-selected options for the AG Attacker Target Select phase.

CheckARMPattern- CheckARMPattern is the module used by AGAttacker to determine if a search profile should be updated for ARM platform flight. ARM platforms are dynamically created. Upon creation, the platform flight is controlled by the user-specified search profile in the ruleset. CheckARMPattern is used to update the flight profile based on new detected target locations.

InitPattern-Calls the routines that dynamically create profile waypoints to fly the AG Attacker in its pattern. InitPattern then sends commands to FP to fly the AG Attacker using the newly created waypoint set.

EndPattern-Completes an AG Attacker pattern profile and removes it from the attacker's profile waypoint list. It also logs the reason for ending the profile in the c3i and post process of log files.

SelectAttackWeapon- SelectAttackWeapon is the module used by AGAttacker to select weapons. For targets with scripted weapons or fire doctrine, the platform will attempt to use those weapons first. If the weapons are not available or no weapons or doctrine has been specified, the weapon with the best PK will be selected. The call to SelectAttackWeapon in Phase 1 simply verifies that a weapon capable against the target is available. This allows the ruleset to transition to Phase 4.

AssignAttackWeapon- Performs part of the weapon selection process. AssignAttackWeapon calls EvaluateAttackWeapon to make weapon/target pairings until the weapon count for the flight has been completed. It handles weapon selection for both wingmen, who can only select their own weapons and for flight leaders, who can select their own and other flight members' weapons.

ProcessCmdTarget- ProcessCmdTarget is the module used by AGAttacker to process the status of a current commanded target. Under certain conditions, the engagement will be ended. The module returns a status indicating: 1) Exit from AGPhase1, 2) Evaluate a weapon for this target, 3) Continue to the next target on the list, or 4) Reschedule AGPhase1.

EndCmdEngage- Terminates a commanded assignment. It generates an Engagement Complete message to send to the commander, deletes the commanded target from the AG Attacker's target list and terminates the profile if the AG Attacker was flying a profile for the commanded target.

CheckAOIPattern- AG Attackers operating in an autonomous role, as part of a SEAD package, will orbit (fly a profile around) its AOI. After the AG Attacker engages a target, it must re-initialize the profile to continue searching the AOI.

AssessAttackTargets

AssessAttackTargets is the module used by AGAttacker to assess detected targets. Detected targets are assessed, prioritized and added to the AGAttacker platform target list for possible engagement.

Module Names & Descriptions for Phase 4

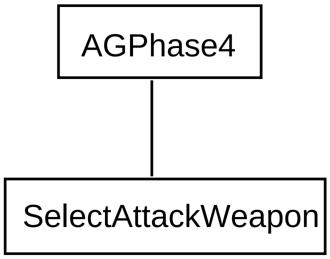

FIGURE 2.11-16. AG Attacker Phase 4 Calling Tree: Lock.

AGPhase4- AGPhase4 is the module that processes the lock phase for the AGAttacker ruleset. During the lock phase, the AGAttacker platform will chose a weapon to engage the target unless the target has a scripted weapon. A launch record is built for each target and the launch phase is scheduled to execute at its start time.

SelectAttackWeapon- SelectAttackWeapon is called from Phase 4 to select the most effective weapon available to engage the target and a launch record is established for it.

Module Names & Descriptions for Phase 5

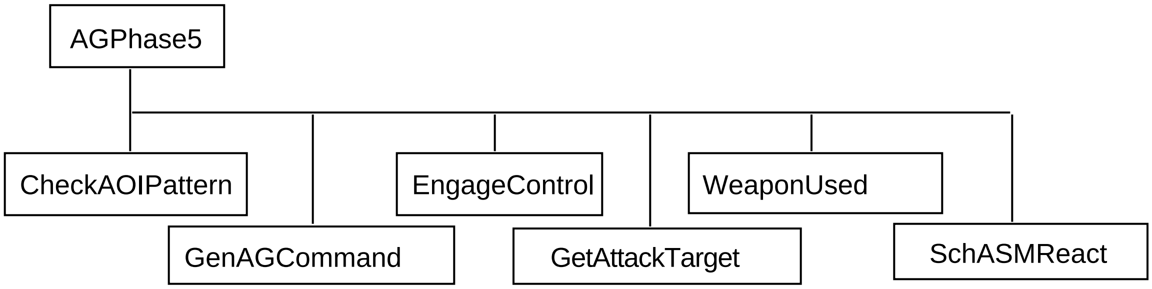

FIGURE 2.11-17. AG Attacker Phase 5 Calling Tree: Launch.

AGPhase5- AGPhase5 is the module that processes the launch phase for the AGAttacker ruleset. The launch phase performs the actual weapon launch.

CheckAOIPattern-In AGPhase5, if a weapon that does not require guidance is launched, the AG Attacker performing in the autonomous role will reinitialize its pattern to search for another target.

WeaponUsed- WeaponUsed is called every time a weapon is launched. The platform's weapon count is decremented. For weapons that can be reloaded, e.g. a Flex SAM ruleset, reload would be initiated.

GetAttackTarget- Checks the target ID and the unique launch ID in the call argument list to find the correct target record and returns the target's record address. The target ID and the launch ID provide a unique identification for the launch phase

Module Names & Descriptions for Phase 6

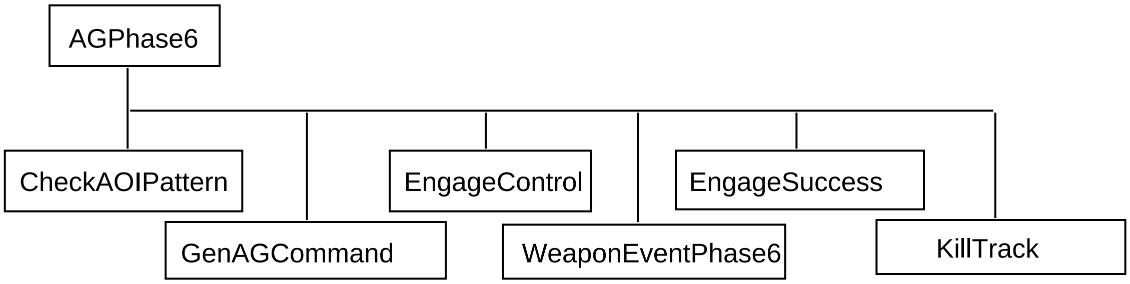

FIGURE 2.11-18. AG Attacker Phase 6 Calling Tree: Intercept.

AGPhase6- AGPhase6 is the module that processes the intercept phase for the AGAttacker ruleset. The intercept phase evaluates the results of the engagement. The evaluation is based on weapon’s lethal radius and probability of kill (PK). If the engagement is judged a success, the intercept will be logged in one of three ways; 1) Fail Dead Ground - weapon hit a dead target, 2) Hit Base - weapon hit an airbase (damage assessment is done at this time), or 3) Success - weapon hit and killed target.

CheckAOIPattern- In AGPhase6, if a weapon that does not require guidance is launched, the AG Attacker performing in the autonomous role will reinitialize its pattern to search for another target.

WeaponEventPhase6- WeaponEventPhase6 is called for cases where the AGAttacker is launching a nuclear weapon to evaluate the success of the detonation.

EngageSuccess- Calls routines that perform the PK draw determines whether the AG Attacker’s engagement succeeded or failed.

KillTrack- AGPhase6 calls KillTrack when it has successfully destroyed a target. KillTrack adjusts the AG Attacker’s track files (general track file, signal track file or ground track file) to indicate the target is dead and should not be subsequently targeted.

Module Names & Descriptions for Phase 7

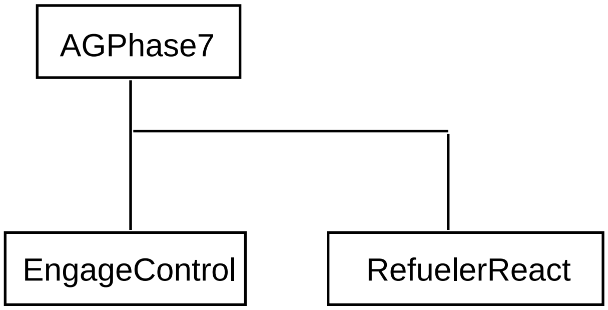

FIGURE 2.11-19. AG Attacker Phase 7 Calling Tree: React to Engage.

AGPhase7- AGPhase7 is the module that processes the React to Engage phase for the AGAttacker ruleset. The AGAttacker React to Engage phase is scheduled in response to an air-to-air engagement against the AGAttacker. This phase is scheduled by an attacker entering the engage mode against the AGAttacker. If the AGAttacker has no subordinates, it attempts to defend itself. This phase has a user-selectable option to disable the reaction. If reaction is disabled, the AGAttacker exits the phase without reacting.

RefuelerReact- RefuelerReact is called when an AG Attacker is engaged or while air refueling is under way. The appropriate reaction phase for the Air Refueler ruleset is scheduled.

Module Names & Descriptions for Phase 8

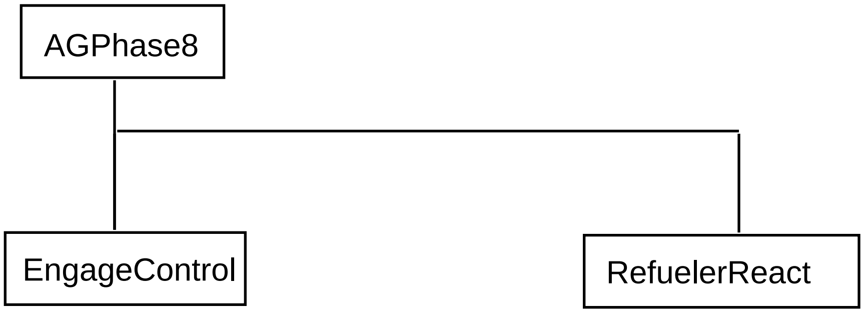

FIGURE 2.11-20. AG Attacker Phase 8 Calling Tree: React to Lock.

AGPhase8- AGPhase8 is the module that processes the React to Lock phase for the AGAttacker ruleset. This phase represents the reaction of the AGAttacker to being locked on by an air-to-air attacker's fire control radar. This phase has a user-selectable option to disable the reaction. If reaction is disabled, the AGAttacker exits the phase without reacting. Weapons specified for jettison are jettisoned, and the AGAttacker performs the Drag action and schedules the drag phase for completion of the maneuver.

Module Names & Descriptions for Phase 9

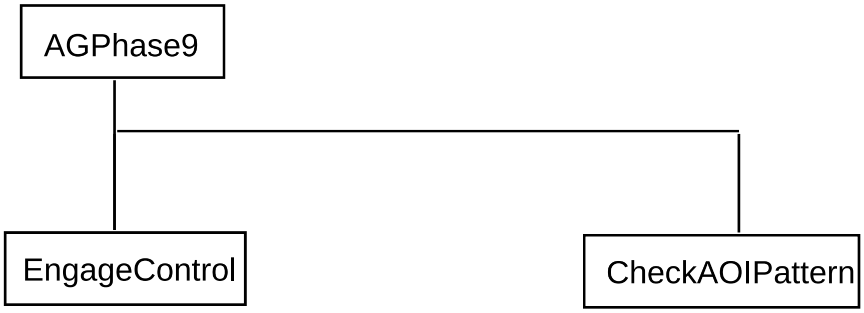

FIGURE 2.11-21. AG Attacker Phase 9 Calling Tree: Drag.

AGPhase9- AGPhase9 is the module that processes the drag phase for the AGAttacker ruleset. The drag maneuver is initiated in AGPhase8. AGPhase9 tests conditions for termination of the drag maneuver. The AGAttacker will continue executing the drag maneuver until no more missiles are in the air to the AGAttacker. Once the drag maneuver has been completed, the platform returns to normal flight.

Module Names & Descriptions for Phase 12

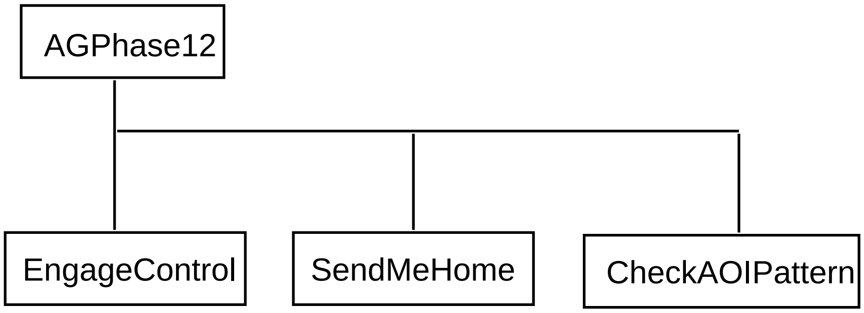

FIGURE 2.11-22. AG Attacker Calling Tree: React to SAM Lock.

AGPhase12- AGPhase12 is the module that processes the react to SAM lock phase for the AGAttacker ruleset. This phase is triggered by a SAM system reaching the decision to engage on the AGAttacker platform. If this reaction is disabled, the AGAttacker exits the phase without reacting.

AG Attacker Phases - State Diagrams

The AG Attacker ruleset consists of eight phases. Each phase is executed over a period of one or more simulation intervals and in each interval either reschedules itself or transitions to a new phase. Figure 2.11-23 is the transition state diagram for the AG Attacker. Phases 1, 4, 5 and 6 are the normal phases in which the AG Attacker operates when it is attacking a target. The AG Attacker transitions to phase 7 (React to Engage) when it has been engaged by a fighter ruleset and to phase 8 (React to Lock) when a fighter has locked on the AG Attacker. Both of these phases are scheduled by the fighter ruleset when it engages or locks on the AG Attacker and are illustrated in Figure 2.11-24. Similarly, phase 12

(React to SAM Lock) is scheduled by a flexible SAM ruleset when it locks on the AG Attacker and is illustrated in Figure 2.11-25.

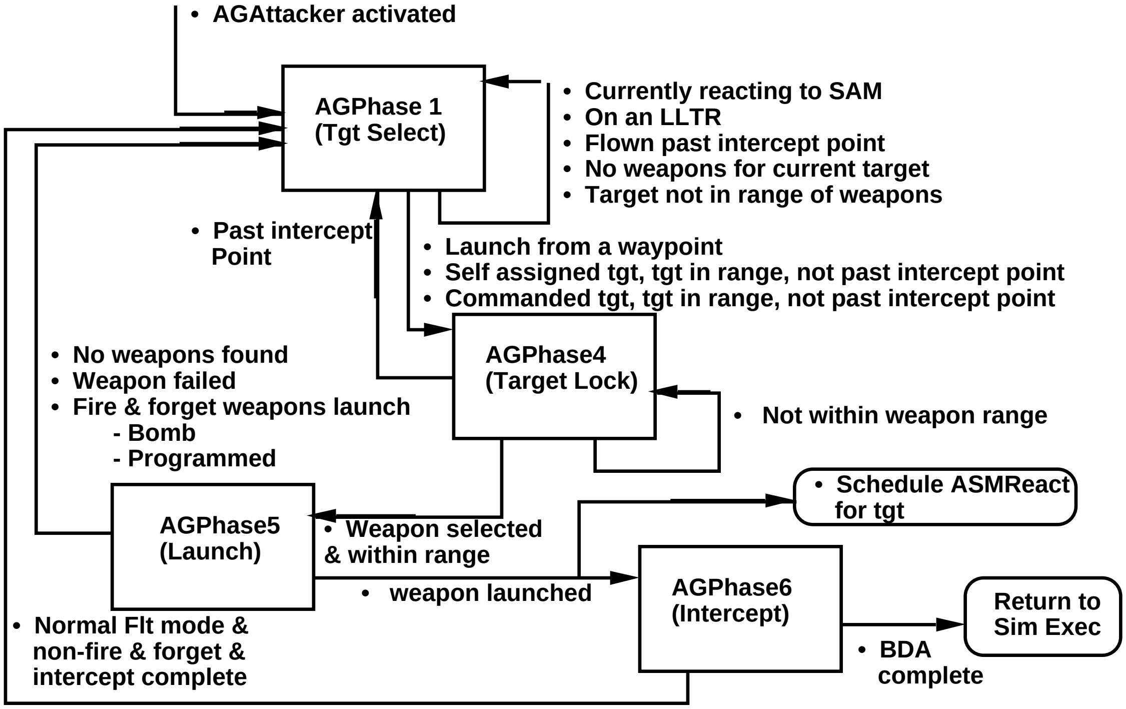
• AGAttacker activated
AGPhase 1
(Tgt Select)
• Currently reacting to SAM
• On an LLTR
• Flown past intercept point
• No weapons for current target
• Target not in range of weapons
• Past intercept Point
• Launch from a waypoint
• Self assigned tgt, tgt in range, not past intercept point
• Commanded tgt, tgt in range, not past intercept point
• No weapons found
• Weapon failed
• Fire & forget weapons launch
- Bomb
- Programmed
AGPhase4
(Target Lock)
• Not within weapon range
• Schedule ASMReact for tgt
AGPhase5
(Launch)
• Weapon selected & within range
• weapon launched
AGPhase6
(Intercept)
Return to Sim Exec
• BDA complete
• Normal Flt mode & non-fire & forget & intercept complete

FIGURE 2.11-23. AG Attacker Target Engagement Phases.

Phase 1, the Target Select Phase, selects targets to be engaged and assigns wingmen with weapons capable against the selected targets. The AG Attacker remains in phase 1, i.e. reschedules AGPhase1:

- If it is reacting to a SAM,
- If it is in a low level transit route and will not exit in the current simulation interval
- If it has flown past the point of expected intercept with the target and must acquire a new target
- If no weapons can be found for the current targets
- If the designated target is not yet within range of the assigned weapons

Phase 1 transitions to the target lock phase, i.e. schedules AGPhase4:

- If the weapon has been designated to launch from a waypoint which has been reached

- If the attacker has self-assigned the target or the attacker has been commanded to engage the target and the target is within range and the attacker has not flown past the expected intercept point

The attacker will remain in phase 4 until it is within engagement range of the assigned weapon type. It will transition to the launch phase, i.e. schedule AGPhase5, when a weapon of the designated type has been selected, has lock on the target and is within launch range. The attacker will transition back to phase 1 if it has flown past the intercept point without achieving lock.

In phase 5 the attacker will either transition to the intercept phase, i.e. schedule AGPhase6, if the weapon has been launched and is semi-active or will transition back to phase 1:

- If no weapons of the required type were available
- If the selected weapon failed to launch
- If the weapon launched and was a fire and forget type

When the attacker schedules phase 6, it also schedules the SAM React phase for the target causing it execute counter measures against the incoming missile.

The attacker will cease to reschedule AGAttacker phases if it has completed Battle Damage Assessment. Otherwise, for normal flight mode, when the non-fire-and-forget intercept is complete, the attacker will transition back to target select mode to find a new target.

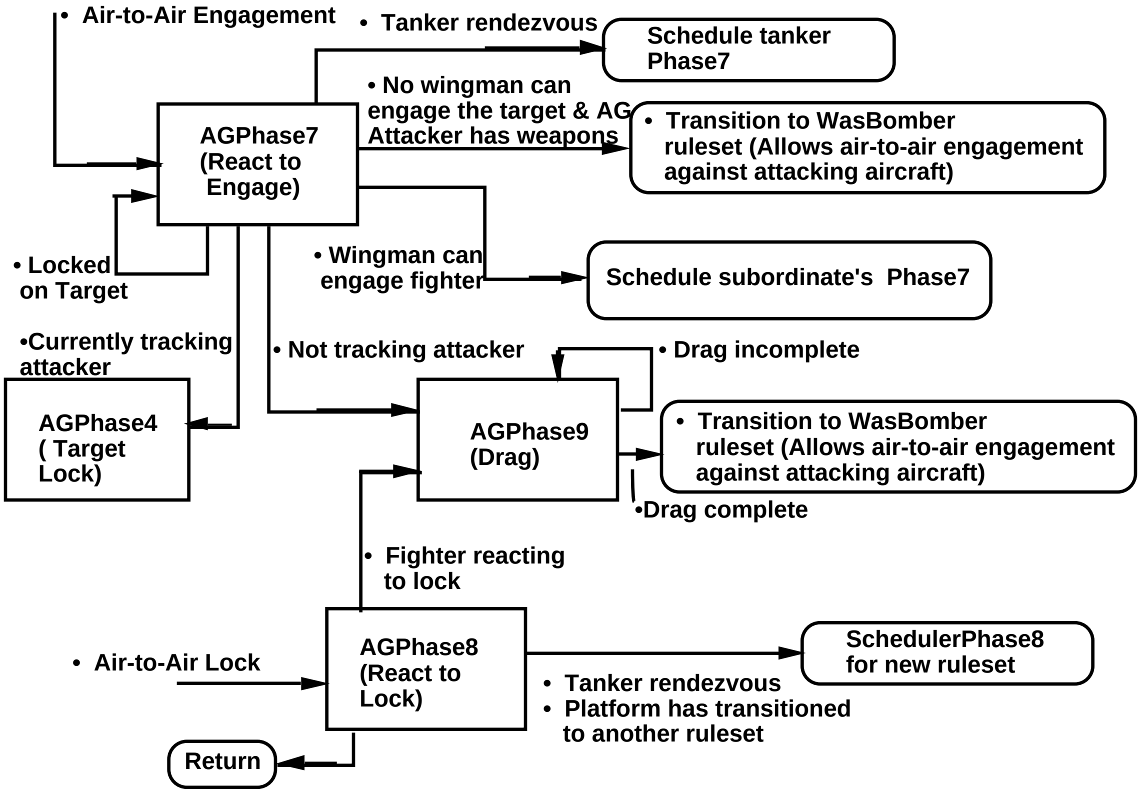
• Air-to-Air Engagement
• Tanker rendezvous
Schedule tanker Phase7
• No wingman can engage the target & AG Attacker has weapons
AGPhase7 (React to Engage)
• Transition to WasBomber ruleset (Allows air-to-air engagement against attacking aircraft)
• Locked on Target
• Wingman can engage fighter
Schedule subordinate's Phase7
•Currently tracking attacker
• Not tracking attacker
• Drag incomplete
AGPhase4 (Target Lock)
AGPhase9 (Drag)
• Transition to WasBomber ruleset (Allows air-to-air engagement against attacking aircraft)
•Drag complete
• Fighter reacting to lock
• Air-to-Air Lock
AGPhase8 (React to Lock)
SchedulerPhase8 for new ruleset
• Tanker rendezvous
• Platform has transitioned to another ruleset
Return

FIGURE 2.11-24. AG Attacker React to Air to Air Phases.

When a fighter engages an AG Attacker, it schedules AGPhase7, React to Engage (Figure 2.11-24). If the AG Attacker is locked on a target, it reschedule phase 7 until launch (for fire and forget interceptor) or intercept (for semi active interceptor). If it is rendezvousing with a tanker, the AG Attacker ruleset schedules the tanker phase 7. If a wingman is available to engage the attacking fighter, its phase 7 is scheduled. If no wingman is available to engage the attacker and the AG Attacker has weapons to do so, it will transition to the Was Bomber ruleset to engage the attacking fighter. If the AG Attacker is currently tracking the attacking fighter, it will schedule AGPhase4 to achieve lock on the target. If the AG Attacker is not tracking the fighter it will schedule AGPhase9 to perform a drag maneuver and, when the maneuver is complete, will transition to the Was Bomber ruleset to engage the fighter.

When a fighter has lock on the AG Attacker, the AG Attacker will schedule AGPhase9 to perform a drag maneuver. If the AG Attacker is scheduled to rendezvous with a tanker or has transitioned to a Was Bomber, phase 8 for the pertinent ruleset is scheduled.

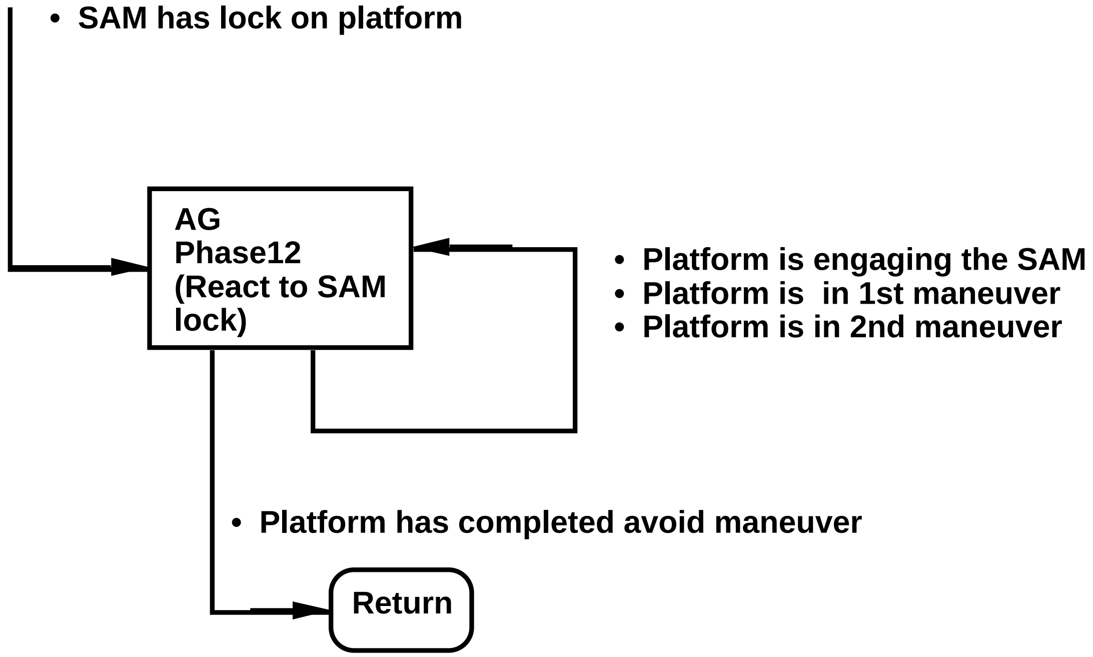
• SAM has lock on platform
AG Phase12 (React to SAM lock)
• Platform is engaging the SAM
• Platform is in 1st maneuver
• Platform is in 2nd maneuver
• Platform has completed avoid maneuver
Return

FIGURE 2.11-25. AG Attacker Phase 12.

When a Flexible SAM platform achieves lock on an AG Attacker, it schedules AGPhase12, React to SAM Lock (Figure 2.11-25), for the AG Attacker. The AG Attacker will remain in phase 12 if it is engaging the SAM and it is in its initial turn maneuver or is in its turn and dive maneuver.

Logic Diagrams and Module Descriptions

AGPhase1

The C3I module, AGPhase1, is the module that processes the target select phase for the AGAttacker ruleset. The inputs and outputs for AGPhase1 are listed in Table 2.11-11 and the flow diagram is illustrated in Figure 2.11-26.

TABLE 2.11-11. Inputs and Outputs for AGPhase1.

Inputs:	
SimTime	Current Simulation Time
Order	Pointer to Engagement Order Struct
Outputs:	
Return Status	Returns 0 (zero) for success

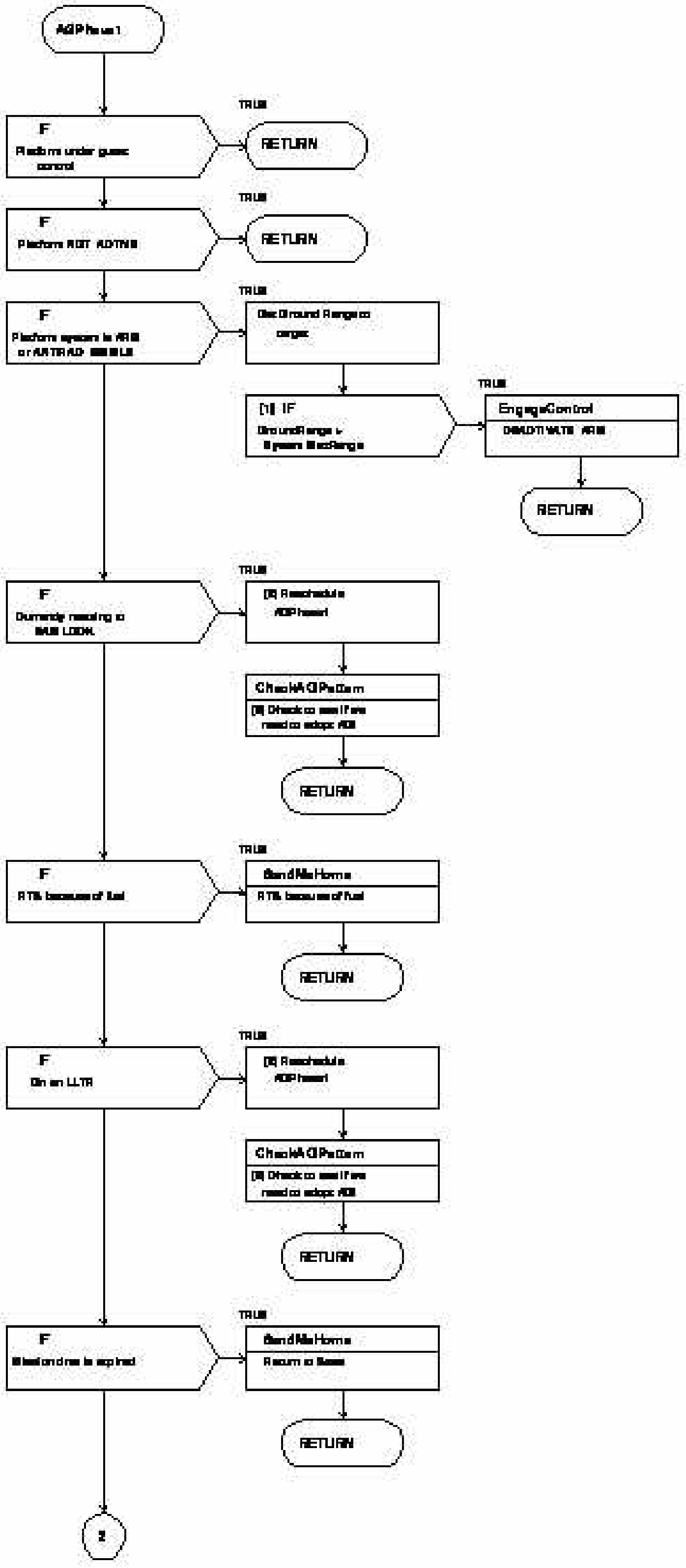

FIGURE 2.11-26. Logic Flow for AGPhase1.

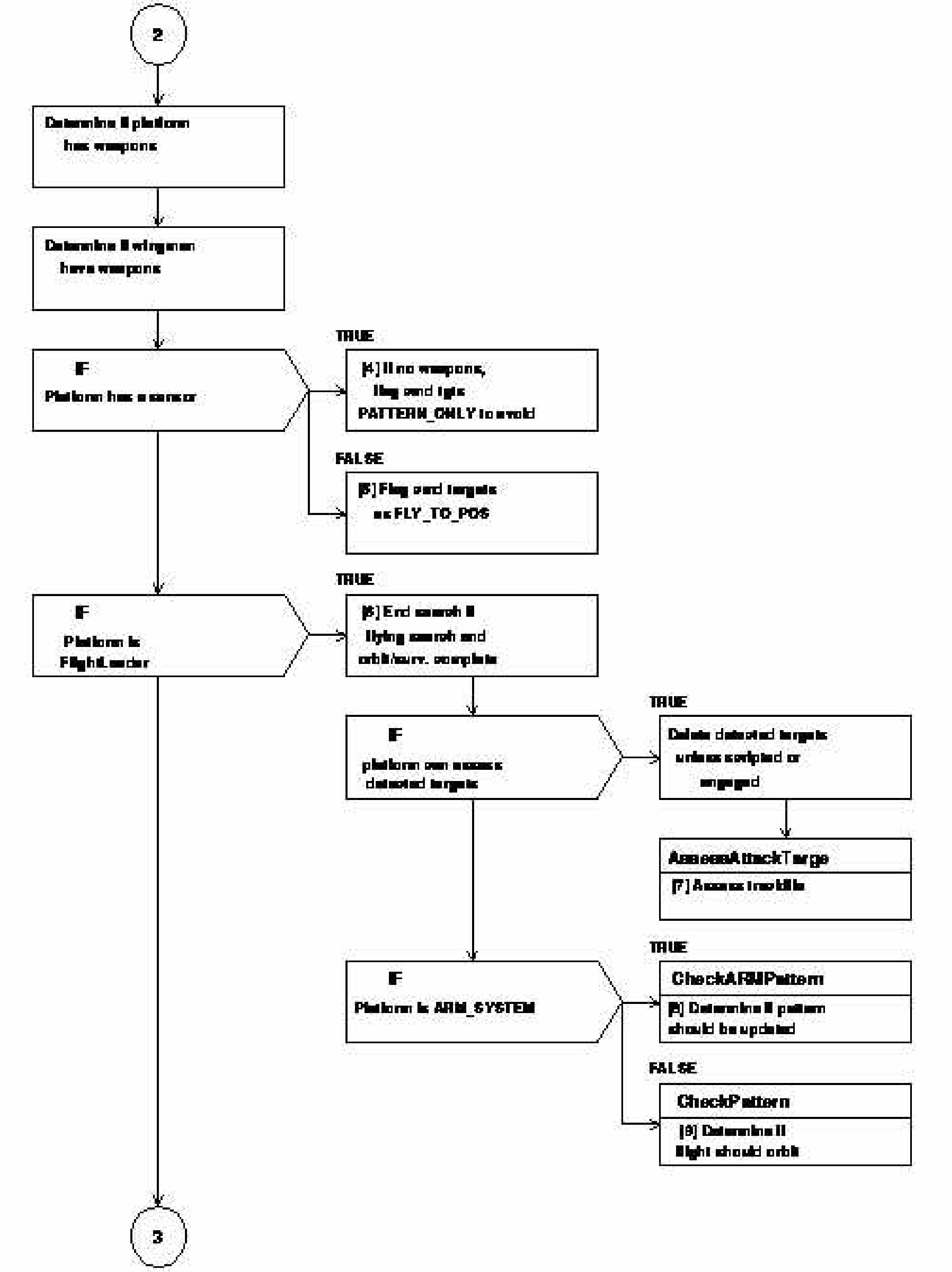
2
Determine if platform has weapons
Determine if wingmen have weapons
IF
Platform has a sensor
TRUE
[4] If no weapons, flag and tgts PATTERN_ONLY to avoid
FALSE
[5] Flag and targets as FLY_TO_POS
IF
Platform is FlightLeader
TRUE
[6] End search if flying search and orbit/surv. complete
IF
platform can assess detected targets
TRUE
Delete detected targets unless scripted or engaged
AssessAttackTarge
[7] Assess Imobile
IF
Platform is ARM_SYSTEM
TRUE
CheckARMPattern
[8] Determine if pattern should be updated
FALSE
CheckPattern
[9] Determine if flight should orbit
3

FIGURE 2.11-26. Logic Flow for AGPhase1. (Contd.)

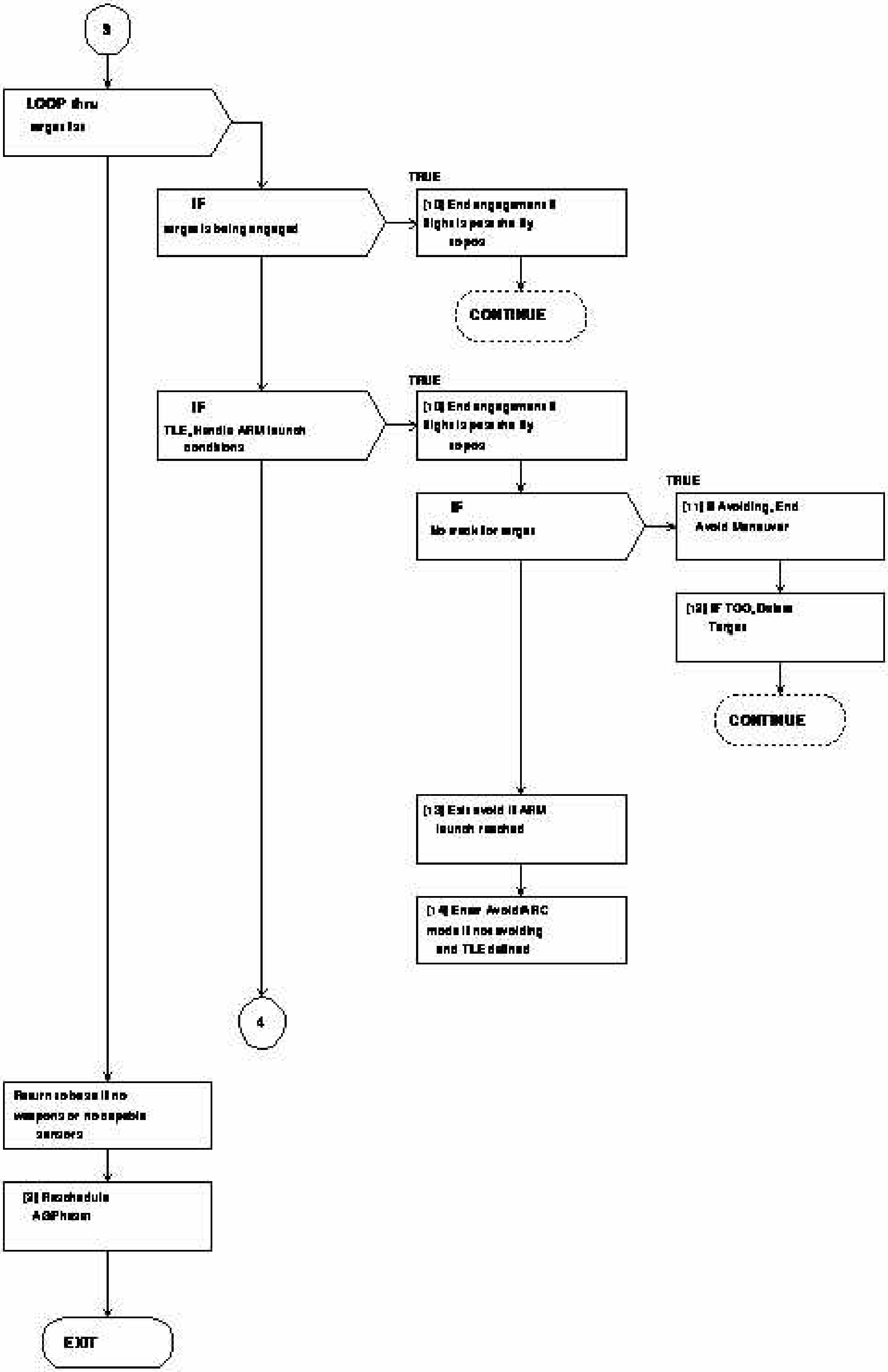

FIGURE 2.11-26. Logic Flow for AGPhase1. (Contd.)

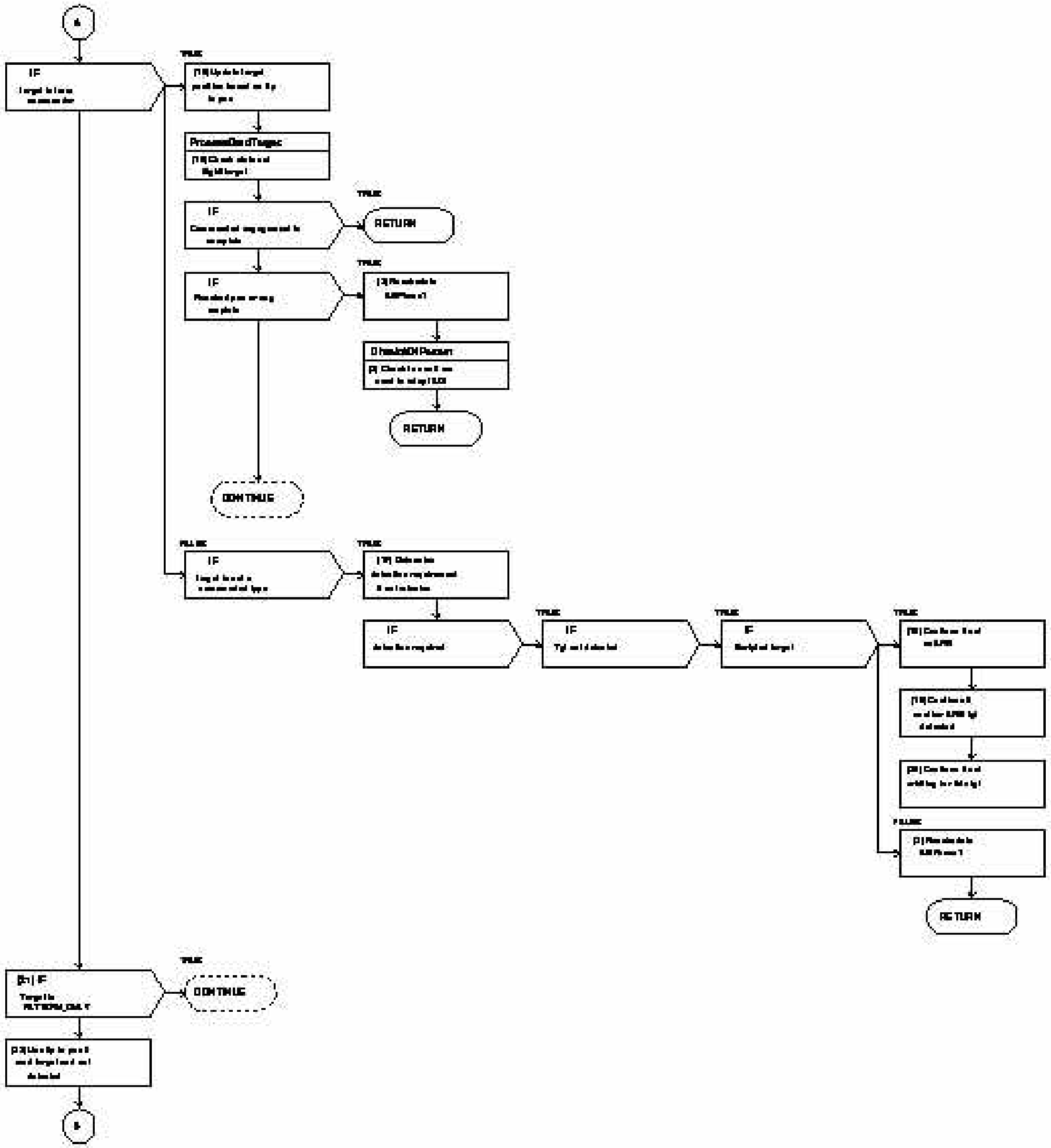

FIGURE 2.11-26. Logic Flow for AGPhase1. (Contd.)

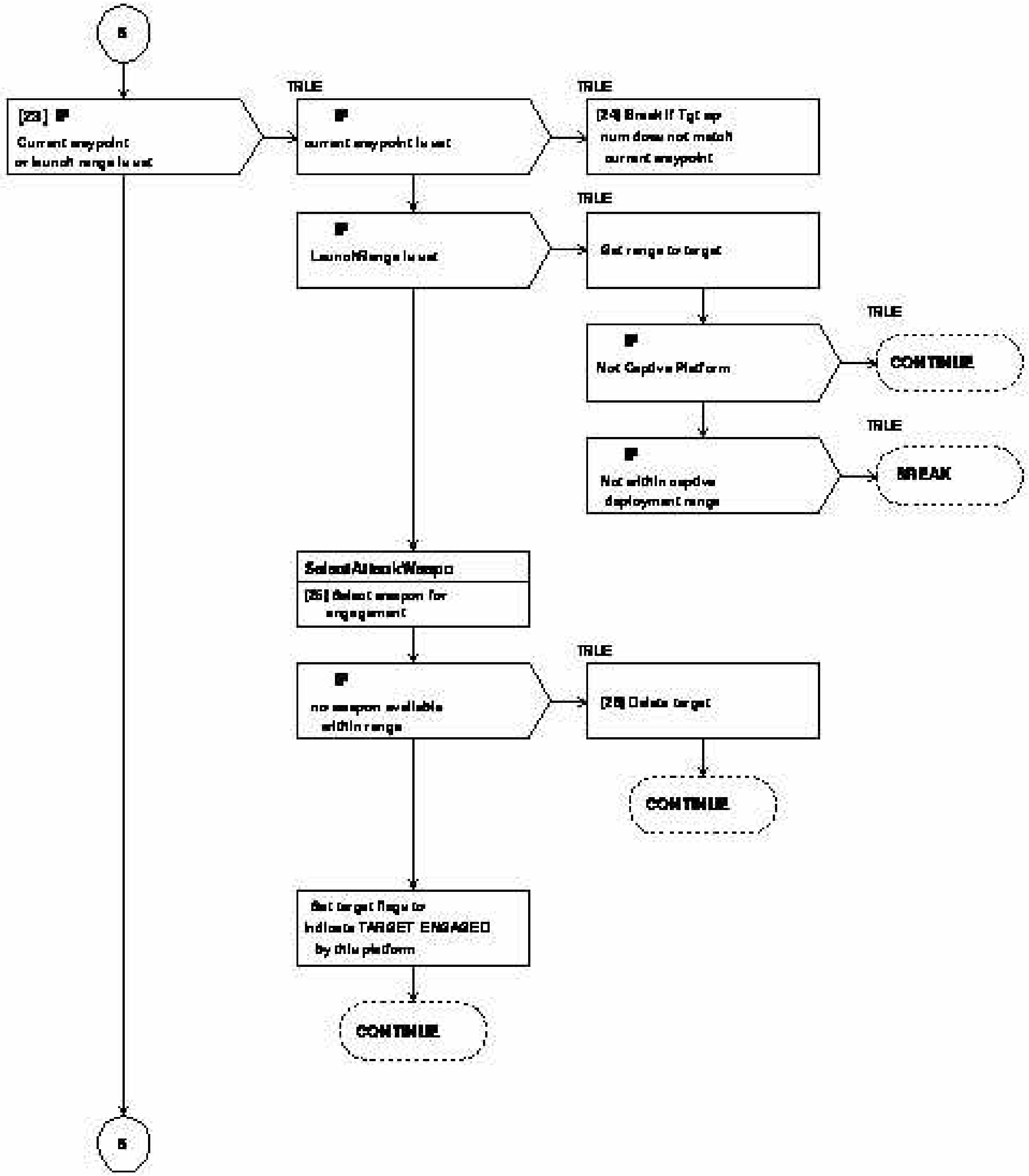

FIGURE 2.11-26. Logic Flow for AGPhase1. (Contd.)

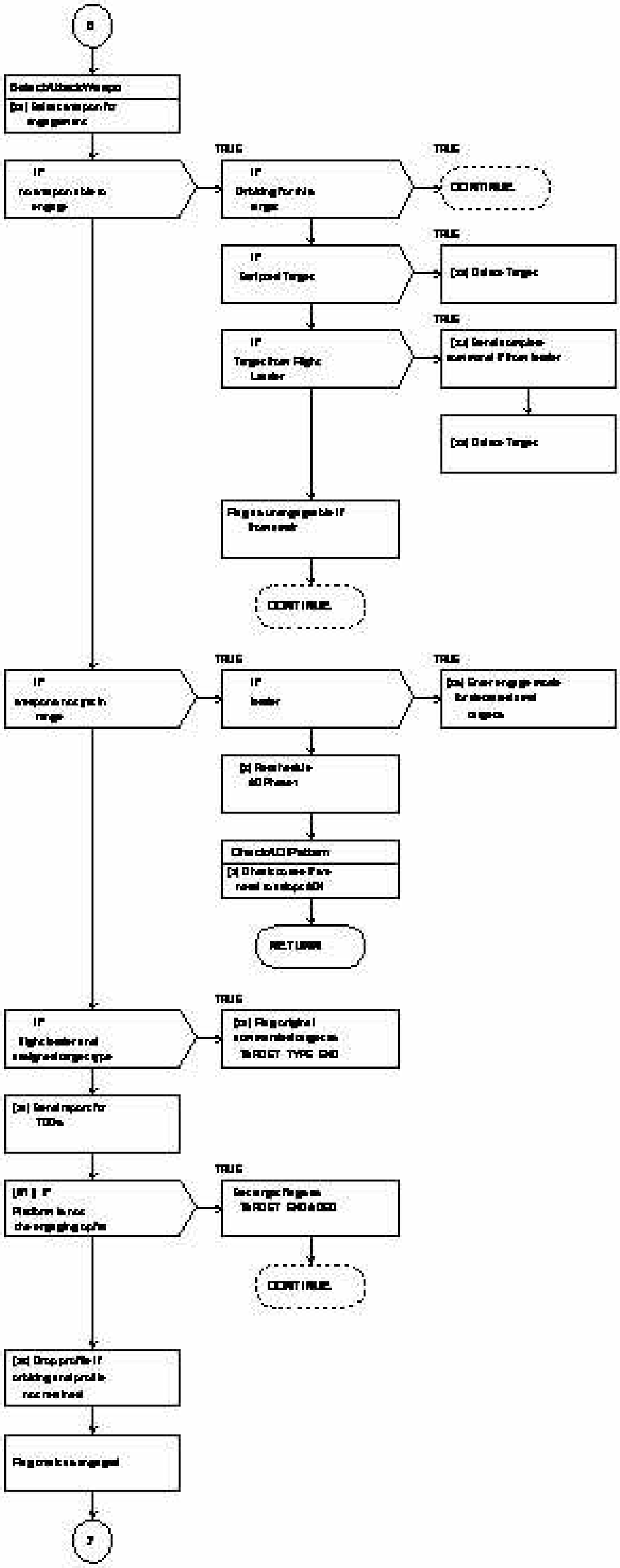
6
SelectAttackWeapon
[ca] Select weapon for engagement
TRUE
IF no weapon able to engage
TRUE
IF Orbiting for this target
TRUE
CONTINUE
TRUE
IF Set pool Target
[ca] Delete Target
TRUE
IF Target from Flight Leader
[ca] Send complete command if from leader
[ca] Delete Target
Flag as unengageable if from self
CONTINUE
TRUE
IF weapons not yet in range
TRUE
IF leader
[ca] Enter engage mode for selected targets
[c] Reschedule AGPhase1
CheckAGPattern
[c] Check to see if we need to adopt AGH
RETURN
TRUE
IF flight leader and assigned target type
[ca] Flag original command target as TARGET TYPE END
[ca] Send report for TODs
TRUE
[IP] IF Platform is not the engaging type
Set target flag as TARGET ENGAGED
CONTINUE
[ca] Drop profile if orbiting and profile not needed
Flag track as engaged
7

FIGURE 2.11-26. Logic Flow for AGPhase1. (Contd.)

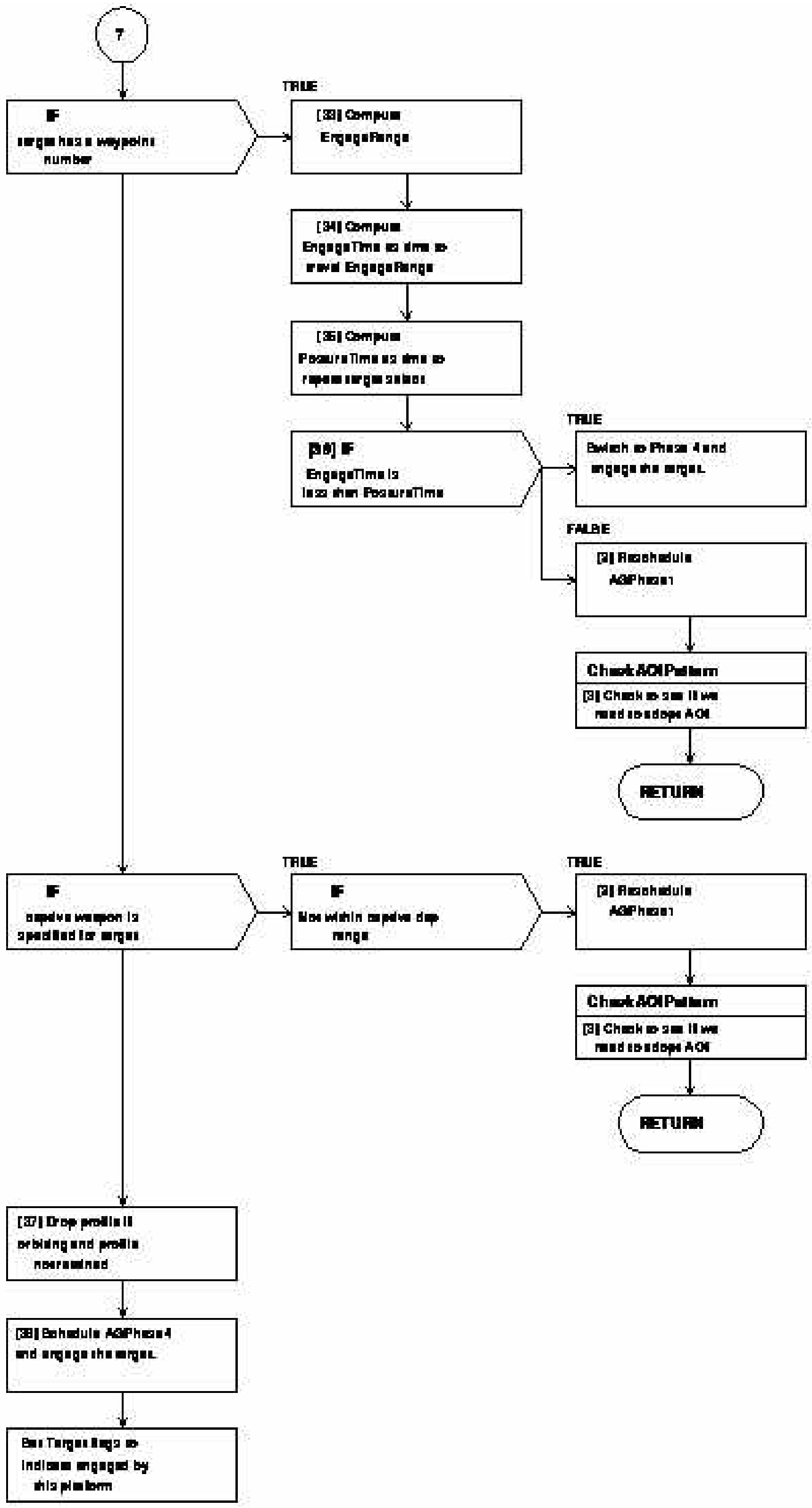
7
TRUE
IF
target has a waypoint number
[33] Compute
EngageRange
[34] Compute
EngageTime as time to travel EngageRange
[35] Compute
PostureTime as time to repeat target select
[36] IF
EngageTime is less than PostureTime
TRUE
Switch to Phase 4 and engage the target.
FALSE
[2] Reschedule AGPhase1
CheckAOIPattern
[3] Check to see if we need to adopt AOI
RETURN
TRUE
IF
captive weapon is specified for target
TRUE
IF
Not within captive drop range
TRUE
[2] Reschedule AGPhase1
CheckAOIPattern
[3] Check to see if we need to adopt AOI
RETURN
[37] Drop profile if orbiting and profile not resumed
[38] Schedule AGPhase4 and engage the target.
Set Target flags to indicate engaged by this platform

FIGURE 2.11-26. Logic Flow for AGPhase1. (Contd.)

Block 1. This block tests for an error condition. The condition is that a weapon using the AGAttacker ruleset was launched at a range beyond the weapon system maximum range capability. If the condition is true, the platform is deactivated (i.e. set inactive for the duration of the scenario).

Block 2. AGPhase1 (i.e. Target Select Phase for AGATTACKER) is rescheduled to execute at current simulation time + the user-specified phase repeat time. For each phase of a ruleset, the user specifies a start time and a repeat time for the phase. These times are delta times from the current simulation time. The start time is the time the phase will execute if scheduled from another phase. The repeat time is the time the phase will execute if it is scheduled from the same phase.

Block 3. CheckAOIPattern is called to determine if an AOI search pattern should be adopted. A pattern will be adopted if the AGAttacker platform is not currently orbiting a target and a target type of TARGET_AOI is found in the platforms target list that is within the user-specified range to initiate a search profile. A target type of TARGET_AOI represents the centroid of a target area or strike package.

Block 4. If the platform has sensors, pointers are established for main and ground track files. If the platform has no weapons, commanded targets will be flagged as PATTERN_ONLY. This is used downstream to avoid executing weapon selection.

Block 5. If the platform has no sensors, then the fly to position provided by the commanded target information is the only position that can be used for engaging the target, therefore, the commanded targets are flagged as FLY_TO_POS.

Block 6. If platform is currently flying a search profile, end the profile if the user-specified number of orbits have been completed. Also turn off surveillance.

Block 7. AssessAttackTargets is called to assess detected targets. Detected targets (i.e. Tracks) are assessed, prioritized and added to the AGAttacker platform target list for possible engagement.

Block 8. CheckARMPattern is called to determine if the ARM platform profile currently being flown should be updated. ARM platforms are basically smart weapons, using the AGAttacker ruleset, that are launched from another platform. ARM platforms are created using the search profile specified in the ruleset. This profile is based on the launch position and the position of the scripted target. If alternate targets are detected, profiles can be updated using the alternate target position.

Block 9. CheckPattern is called to determine if a search or track profile should be initiated for the AGAttacker platform flight. Search and Track profiles are specified by the user in the ruleset. Initiation of a profile is based on the platform being within profile initiation range of the target.

Block 10. If platform has flown past the commanded fly-to position for the target, end the engagement and process the next target in the target list.

Block 11. If the AGAttacker no longer has track data for target, go back to normal flight.

Block 12. If the AGAttacker no longer has track data for target and the target is a target of opportunity (TOO), remove the target from the platform's target list.

Block 13. If still tracking target and ARM launch conditions have been met, exit the avoid maneuver if avoiding.

Block 14. If still tracking target and ARM launch conditions have not been met and user has specified requirements for time in track before ARM launch, enter avoid/arc maneuver. This will allow target to stay in track until ARM launch conditions have been met.

Block 15. Get target position from the platform's commanded fly-to position. This will be passed in to ProcessCmdTarget.

Block 16. ProcessCmdTarget is called to evaluate conditions regarding a commanded target. The return values indicate a need to: 1) exit from target select phase, 2) evaluate a weapon for this target, 3) continue to next target, or 4) reschedule target select.

Block 17. Determine target identification requirement (i.e. by class or by system type). Target system/class is determined and then ruleset specifies detection requirement for system/class.

Block 18. Continue to next target if AGAttacker is not setup as an ARM.

Block 19. Continue to next target if another ARM target is detected.

Block 20. Continue to next target if not orbiting this target.

Block 21. Continue to next target if this target is marked as PATTERN_ONLY.

Block 22. Use Fly To Pos for target position if target has been commanded but not detected.

Block 23. If CurrentWaypoint has been set, this indicates a weapon release at a particular waypoint number. If LaunchRange has been set, this indicates that deployment range has been met for launching captive platforms.

Block 24. If the target waypoint number does not match the current waypoint, break out AGPhase1.

Block 25. Select AttackWeapon is called to select the best weapon for engaging the target. For targets with scripted weapons, the platform will attempt to use those weapons first. If scripted weapons are not available or no weapons or doctrine has been specified, the weapon with the best probability of kill (PK) will be selected.

Block 26. Target is removed from platform target list.

Block 27. Command Complete message is sent to the flight leader indicating platform can not comply (CANTCO).

Block 28. Platform transitions to engage mode and flies directly toward target.

Block 29. Indicate target as engaged in target structure.

Block 30. Send out command reports for targets of opportunity indicating flight leader is the engager.

Block 31. Wingman is the engaging platform.

Block 32. If flight leader is flying search or track profile, and ruleset does not specify retain profile, end the profile in order to engage the target.

Block 33. Compute engage range, i.e. Range from current platform position to weapon release waypoint.

Block 34. Compute EngageTime, i.e. Range/Velocity.

Block 35. Compute PostureTime. This is the time required repeat target select phase.

Block 36. If there is not enough time to repeat the target select phase, transition to the lock phase and engage the target.

Block 37. If platform is flying search or track profile, and ruleset does not specify retain profile, end the profile in order to engage the target.

Block 38. Schedule the lock phase for AGAttacker and engage the target.

AGPhase4

The C3I module, AGPhase4, is the module that processes the lock phase for the AGAttacker ruleset. The inputs and outputs for AGPhase4 are listed in Table 2.11-12 and the steps performed by the module are shown in Figure 2.11-27.

TABLE 2.11-12. Inputs and Outputs for AGPhase4.

Inputs:	
SimTime	Current Simulation Time
Order	Pointer to Engagement Order Struct
Outputs:	
Return Status	Returns 0 (zero) for success

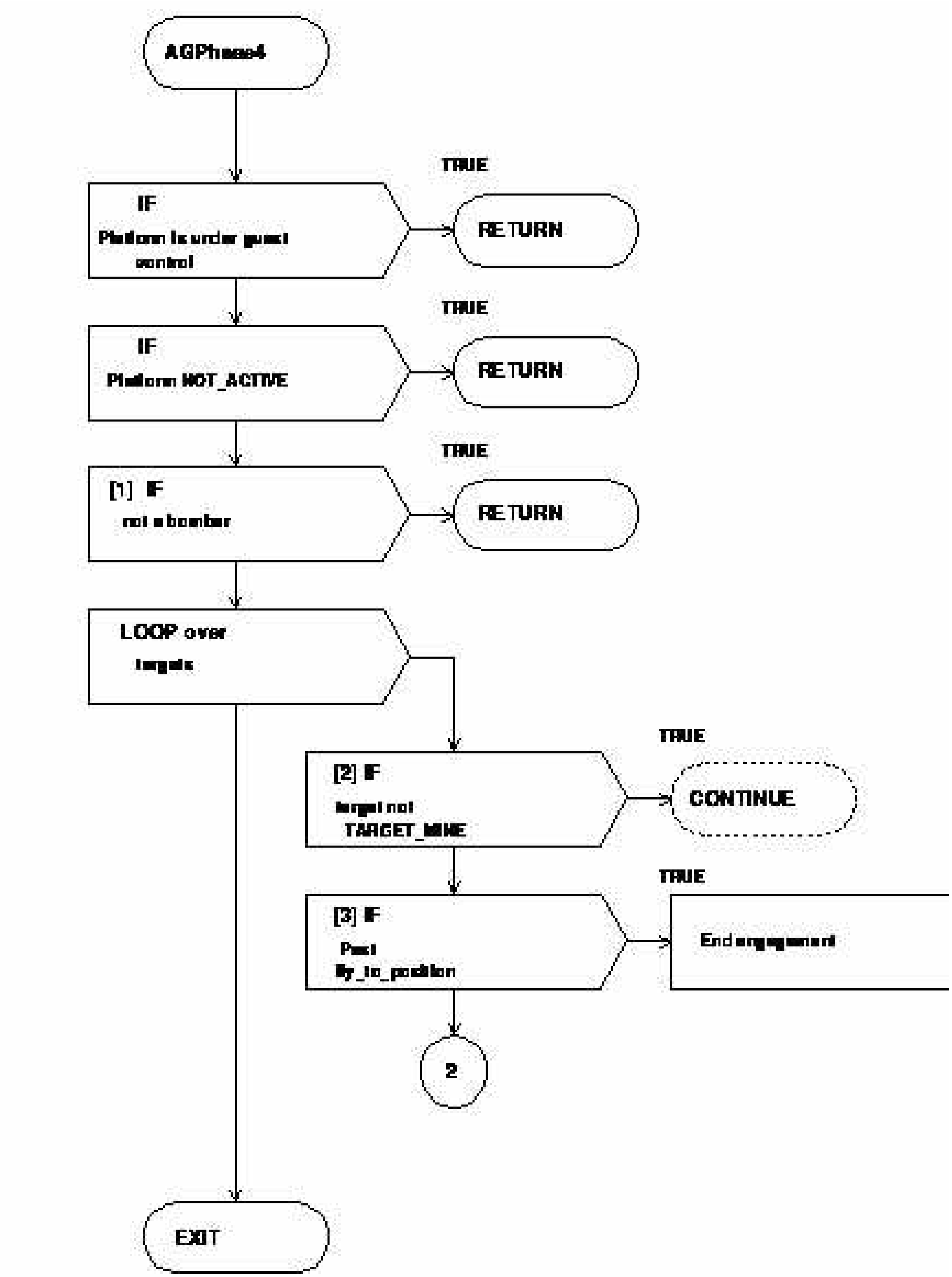

FIGURE 2.11-27. Flow Diagram for AGPhase 4: Lock.

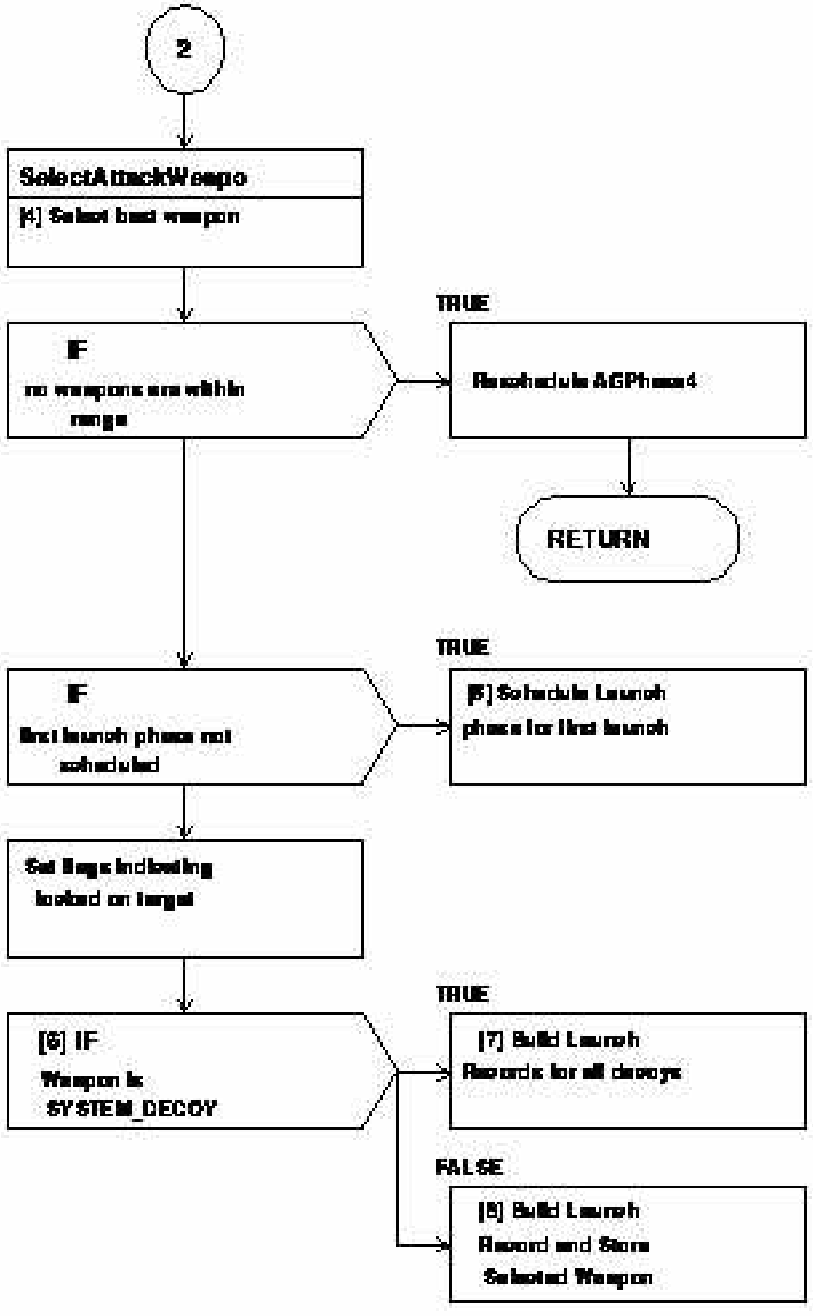
2
SelectAttackWeapo
[4] Select best weapon
IF
no weapons are within range
TRUE
Reschedule AGPhase4
RETURN
IF
first launch phase not scheduled
TRUE
[5] Schedule Launch phase for first launch
Set flags indicating locked on target
[6] IF
Weapon is SYSTEM_DECOY
TRUE
[7] Build Launch Records for all decoys
FALSE
[8] Build Launch Record and Store Selected Weapon

FIGURE 2.11-27. Flow Diagram for AGPhase 4: Lock. (Contd.)

Block 1. This block tests for the conditions that the AGAttacker has transitioned to the WAS BOMBER ruleset. Transition to the WAS BOMBER ruleset can occur during a reaction to being engaged or at the completion of a drag maneuver.

Block 2. The condition is tested that the target is mine (TARGET_MINE). This indicates that I am the engaging platform. If I am not the engaging platform, continue to the next target.

Block 3. Flight leaders check to see if they have past the fly_to_position for commanded targets, if so, the commanded engagement is stopped and we return out of AGPhase4.

Block 4. Select AttackWeapon is called to select the best weapon for engaging the target. For targets with scripted weapons, the platform will attempt to use those weapons first. If scripted weapons are not available or no weapons or doctrine has been specified, the weapon with the best probability of kill (PK) will be selected.

Block 5. Weapons are available and within range. Launch phase (AGPhase5) is scheduled at its start time to handle launching weapon.

Block 6. This test determines if the selected weapon type is SYSTEM_DECOY. SYSTEM_DECOY weapon launches are handled differently from other weapon types.

Block 7. If the weapon selected for launch is marked as SYSTEM_DECOY, a separate launch record is built for each decoy (i.e. as many as is defined by the weapon count for the decoy weapon).

Block 8. If the weapon selected for launch is not marked as SYSTEM_DECOY, a single launch record is built for the weapon.

AGPhase5

The C3I module, AGPhase5, is the module that processes the launch phase for the AGAttacker ruleset. The inputs and outputs for AGPhase5 are listed in Table 2.11-13 and the steps performed by the module are shown in Figure 2.11-28.

TABLE 2.11-13. Inputs and Outputs for AGPhase5.

Inputs:	
SimTime	Current Simulation Time
Order	Pointer to Engagement Order Struct
Outputs:	
Return Status	Returns 0 (zero) for success

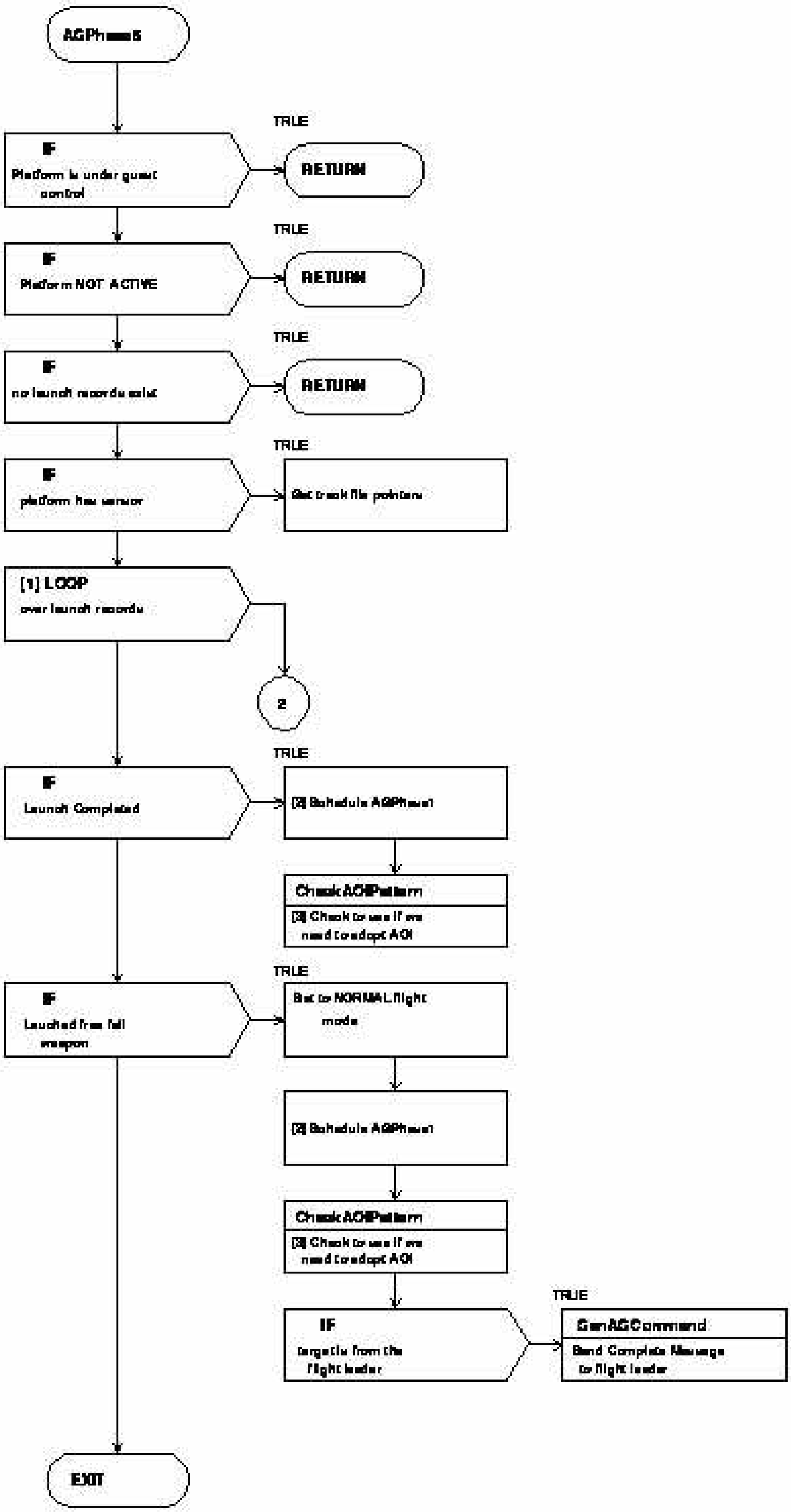

FIGURE 2.11-28. Flow Diagram for AGPhase 5: Launch.

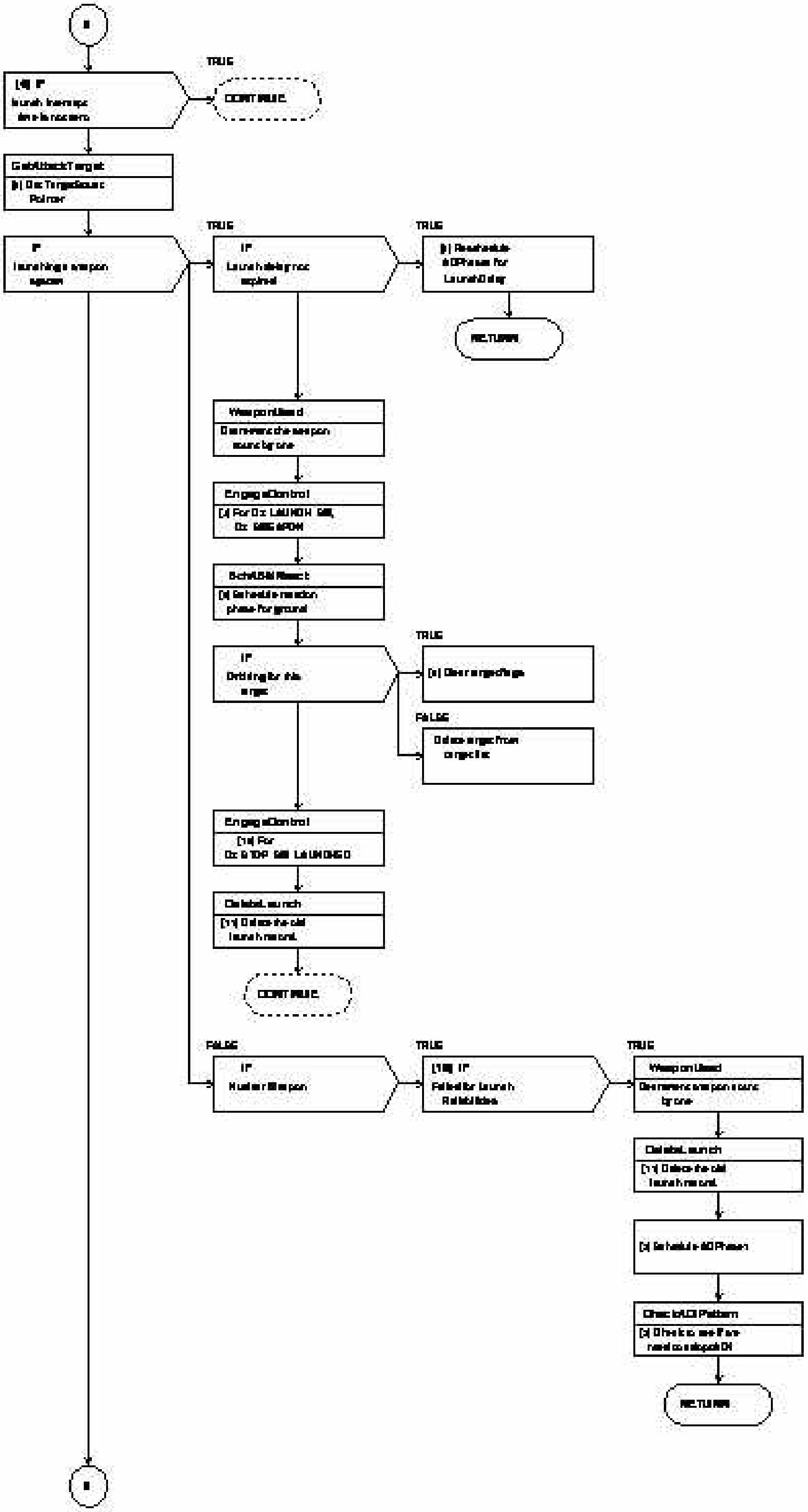

FIGURE 2.11-28. Flow Diagram for AGPhase 5: Launch. (Contd.)

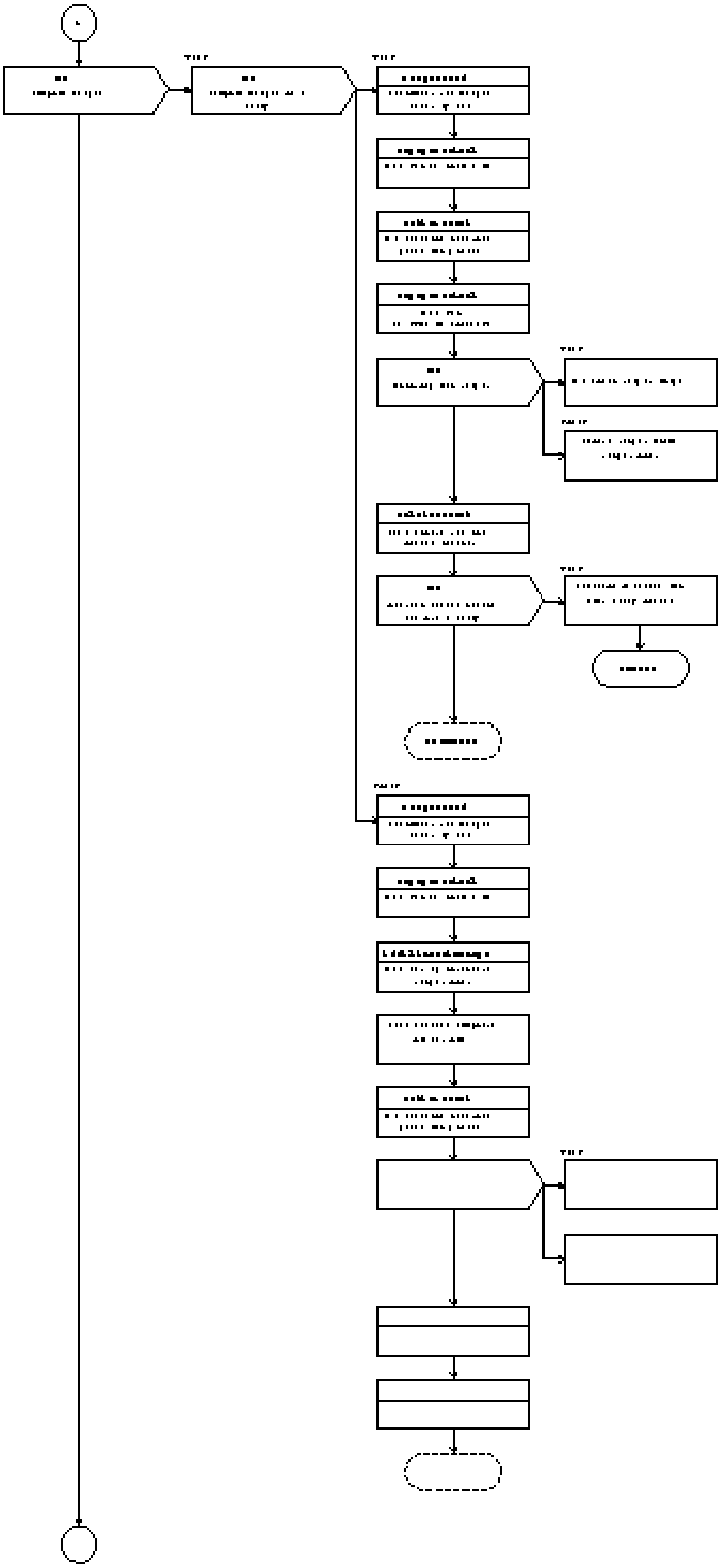

FIGURE 2.11-28. Flow Diagram for AGPhase 5: Launch. (Contd.)

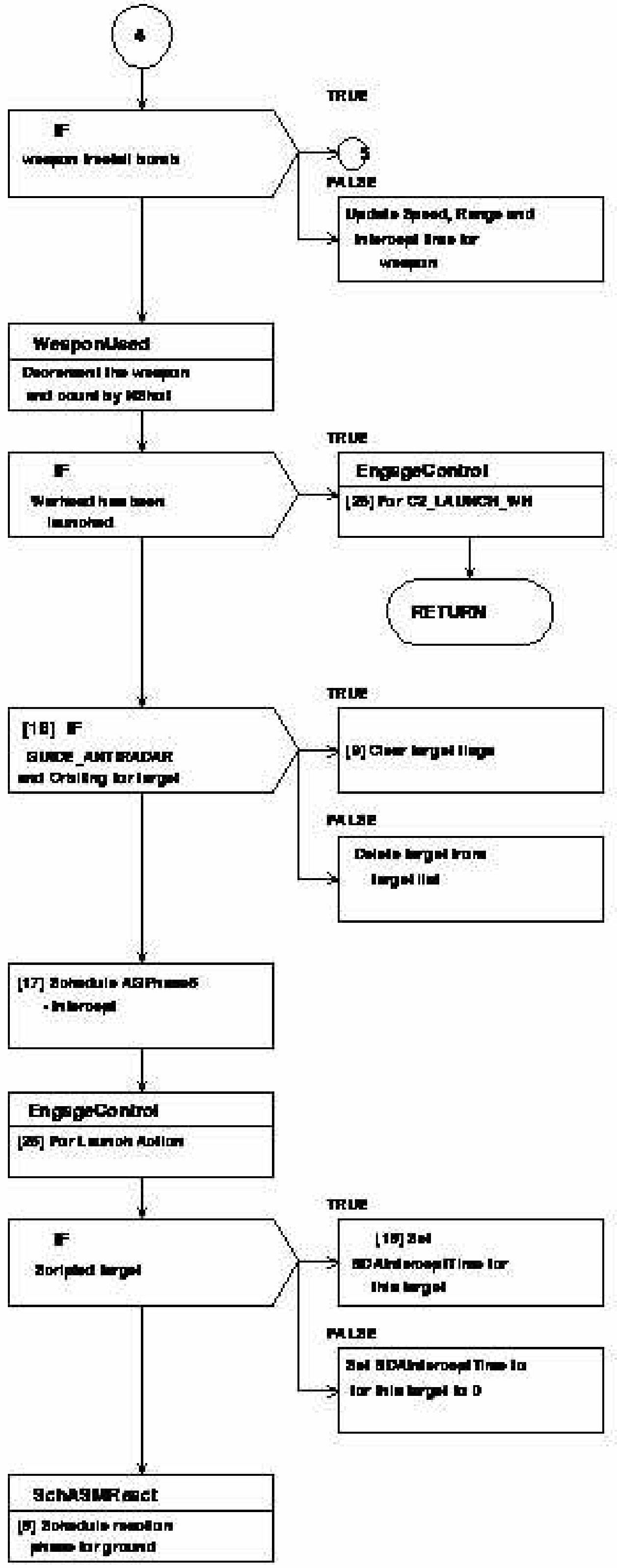

FIGURE 2.11-28. Flow Diagram for AGPhase 5: Launch. (Contd.)

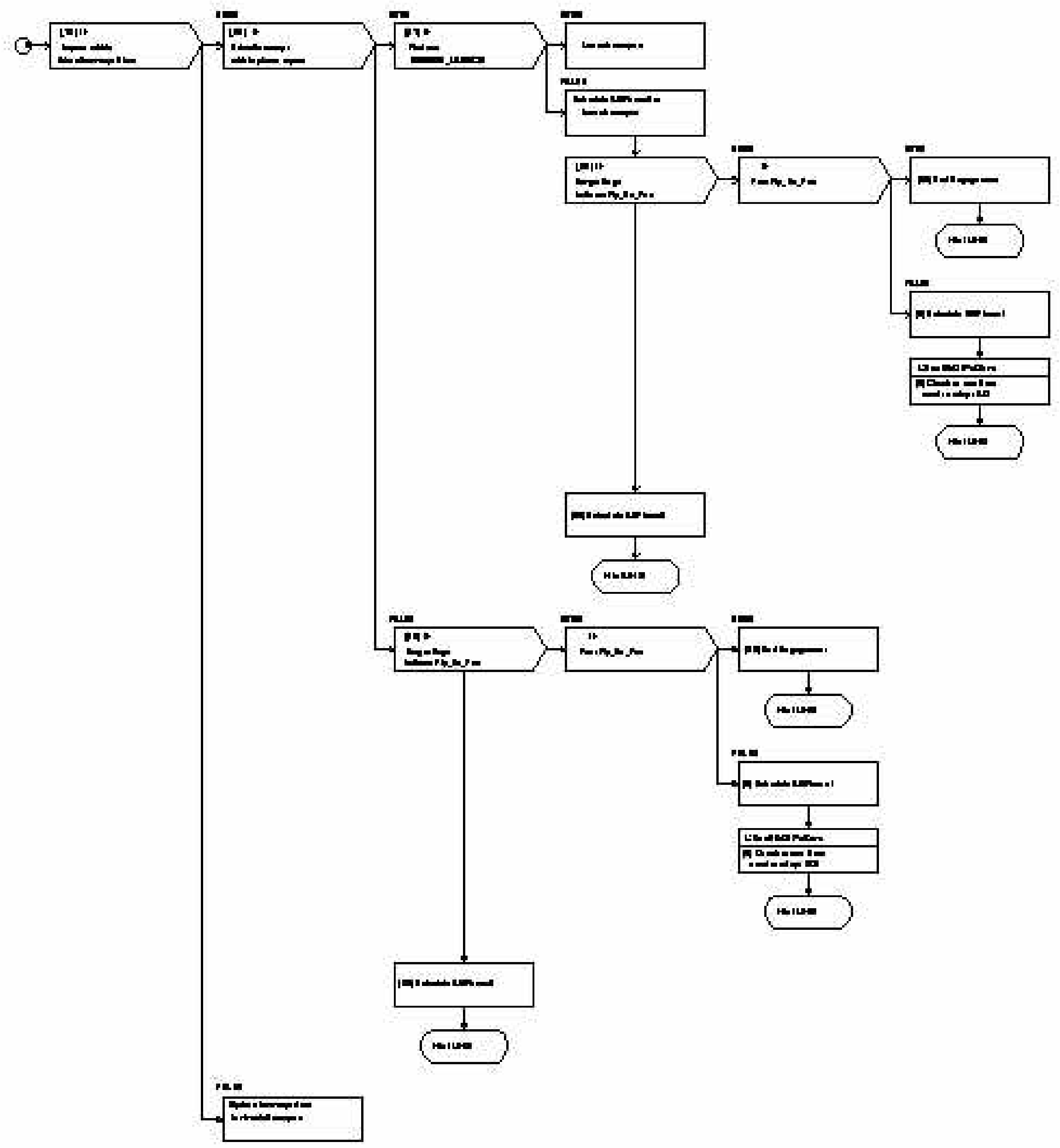

FIGURE 2.11-28. Flow Diagram for AGPhase 5: Launch. (Contd.)

Block 1. Loop over all launch records for the platform.

Block 2. AGPhase1 (i.e. Target Select Phase for AGATTACKER) is rescheduled to execute at current simulation time + the user-specified phase repeat time.

Block 3. CheckAOIPattern is called to determine if an AOI search pattern should be adopted. A pattern will be adopted if the AGAttacker platform is not currently orbiting a target and a target type of TARGET_AOI is found in the platforms target list that is within the user-specified range to initiate a search profile. A target type of TARGET_AOI represents the centroid of a target area or strike package.

Block 4. If the Launch->IntTime is non-zero, this means we have already processed this launch record, go to the next one.

Block 5. GetAttackTarget is called to get the target structure pointer for the targetID (i.e. the target being launched on) specified in the launch record being processed.

Block 6. AGPhase5 (i.e. Launch phase for AGAttacker) is rescheduled at current simulation time + launch delay.

Block 7. EngageControl is called to log a C2_LAUNCH_SW action (i.e. record what weapon was launched and how many), and to send a C2_SWEAPON to flight processing to activate a captive platform.

Block 8. SchASMReact is called to schedule the reaction phase for the ground platform that we are launching a weapon against.

Block 9. The TARGET_ENGAGED, TARGET_MINE flags are cleared and the TARGET_PROF_ENGAGED is set to indicate engaging from a profile. This flag will be used downstream to indicate that the target was detected and an interceptor launched against it from a profile.

Block 10. EngageControl is called to record a C2_STOP_SW_LAUNCHED action (i.e. stop smart weapon launched). A stop message is sent to flight processing. The launching AGAttacker platform transitions to normal flight mode (i.e. flying its user-specified waypoints) and is no longer engaging the target. After a smart weapon has been launched, the outcome of the engagement will be evaluated by the smart weapon. The launching AGAttacker's target select phase can be rescheduled after the launch.

Block 11. The launch record for the weapon that was just launched is no longer needed, so it is deleted/deallocated.

Block 12. The condition is checked to determine if a random draw is less than the user specified weapon reliability value.

Block 13. EngageControl is called to log a C2_LAUNCH_SW action (i.e. record what weapon was launched and how many). A launch message is sent to flight processing.

Block 14. For non-decoy complex weapons, AddAlternateTargets is called to add any alternate targets within the field of view of the newly created complex weapon platform that passes the systems of interest list and AOI checks.

Block 16. The conditions checked in this block indicate that the AGAttacker launching platform was orbiting this target and a simple weapon with guidance type of antiradar was launched.

Block 17. AGPhase6 (i.e. Intercept Phase for AGAttacker) is scheduled at its start time. The intercept (kill/no kill) will be evaluated in this phase.

Block 18. If the target is a scripted target, the battle damage assessment (BDA) intercept time is set to indicate a BDA should be performed by an intel center. Otherwise, the BDA intercept time is set to zero indicating the intel centers will not perform BDA.

Block 19. A flat earth approximation is used to compute the time from launch to impact. The time required for the weapon to fall to the target's altitude is computed based on the initial velocity (i.e. the velocity of the launching AGAttacker platform) and the force of gravity acting on the weapon. The condition checked in this block is: can the weapon traverse the distance in the z direction from attacker position to target position in the interval specified by (impact time - launch time).

Block 20. If the weapon can reach the target altitude within the current launch phase repeat time the launch continues.

Block 21. If the platform flags indicate it is still ok to launch, the weapon is launched. Otherwise, AGPhase5 is rescheduled at its repeat time.

Block 22. If this is a commanded target and platform has past the commanded fly to position, the engagement will be ended.

Block 23. AGPhase5 (Launch Phase for the AGAttacker) is rescheduled at its repeat time.

Block 24. Having flown past the fly to pos provided for the commanded target, the AGAttacker ends the engagement.

Block 25. EngageControl is called to log a C2_LAUNCH action (i.e. record what weapon was launched and how many). A launch message is sent to flight processing.

Block 26. EngageControl is called for the launch warhead action.

AGPhase6

The C3I module, AGPhase6, is the module that processes the intercept phase for the AGAttacker ruleset. Inputs and outputs for AGPhase6 are listed in Table 2.11-14 and the steps performed by the module are shown in Figure 2.11-29.

TABLE 2.11-14. Inputs and Outputs for AGPhase6.

Inputs:	
SimTime	Current Simulation Time
Order	Pointer to Engagement Order Struct
Outputs:	
Return Status	Returns 0 (zero) for success

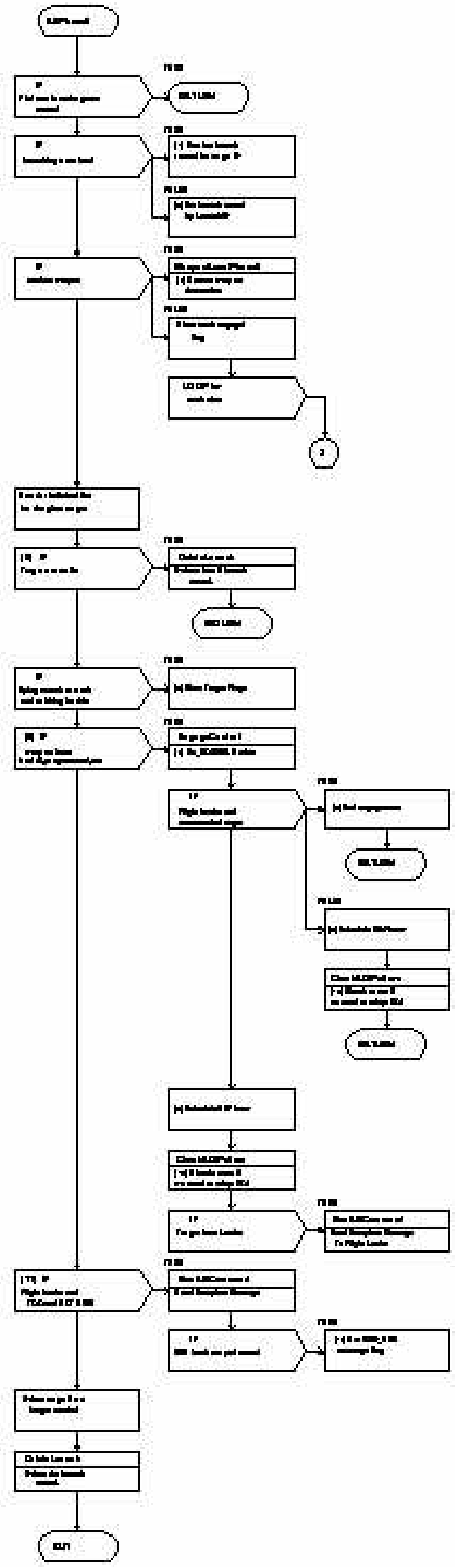

FIGURE 2.11-29. Flow Diagram for AGPhase 6: Intercept.

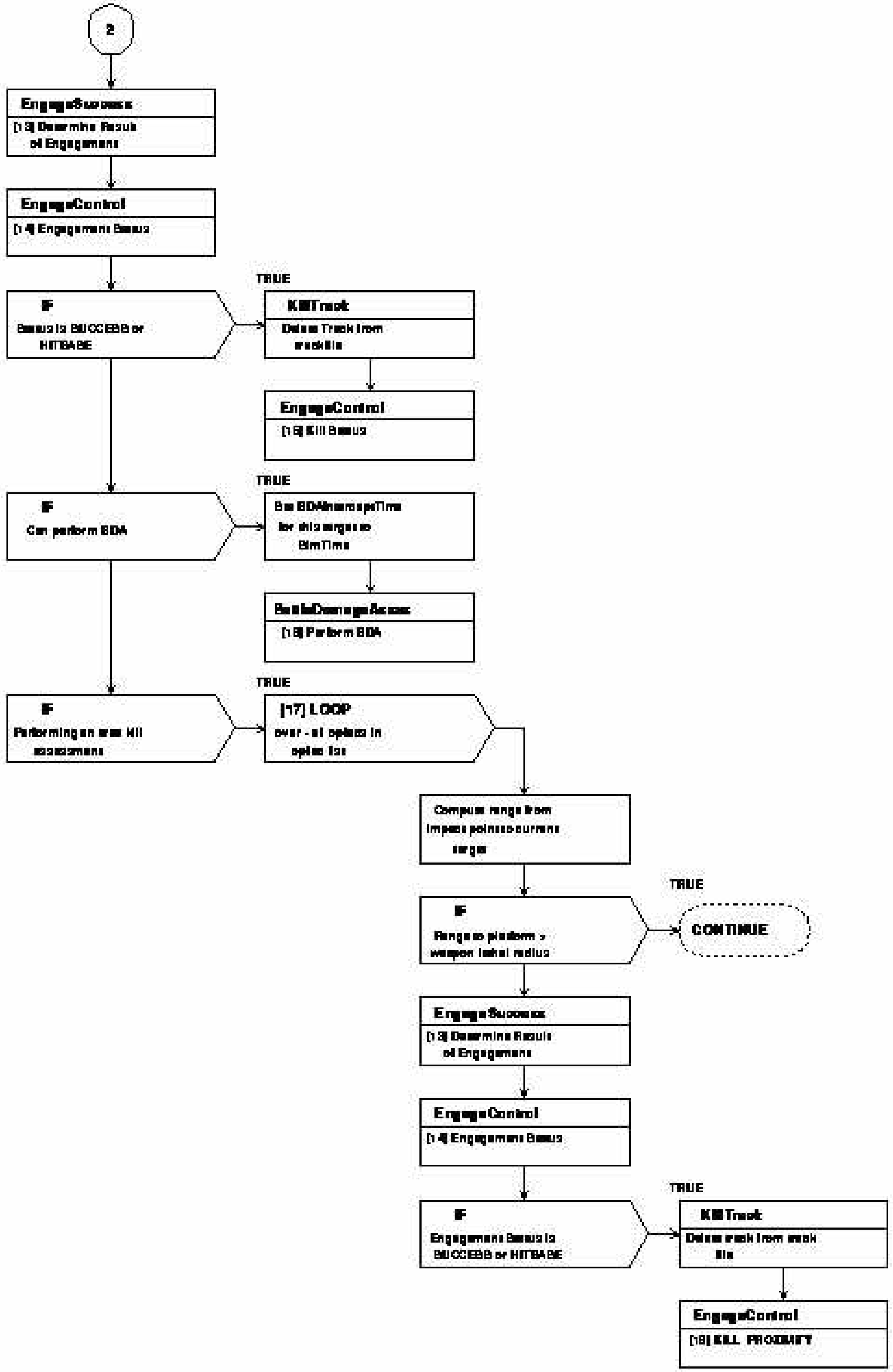
2
EngageSuccess
[13] Determine Result of Engagement
EngageControl
[14] Engagement Status
IF
Status is SUCCESS or HITBASE
TRUE
KillTrack
Delete Track from tracklist
EngageControl
[15] Kill Status
IF
Can perform BDA
TRUE
Set BDAInterceptTime for this target to SimTime
BattleDamageAssess
[16] Perform BDA
IF
Performing an area kill assessment
TRUE
[17] LOOP
over - all options in option list
Compute range from impact point to current target
IF
Range to platform > weapon lethal radius
TRUE
CONTINUE
EngageSuccess
[13] Determine Result of Engagement
EngageControl
[14] Engagement Status
IF
Engagement Status is SUCCESS or HITBASE
TRUE
KillTrack
Delete track from track file
EngageControl
[18] KILL PROXIMITY

FIGURE 2.11-29. Flow Diagram for AGPhase 6: Intercept. (Contd.)

Block 1. For warhead weapons, the last launch record for this TargetID is used to evaluate the result of the engagement. If launching a warhead, the launch record associated to the warhead weapon will always be the last launch record in the launch linked list.

Block 2. WeaponEventPhase6 is called to evaluate nuclear weapon detonation effects.

Block 3. If the target related to this launch is not found, the target has been reassigned, therefore, the launch record is no longer needed.

Block 4. If the AGAttacker is flying a profile (search or track), the target flags TARGET_ENGAGED and TARGET_MINE are cleared and the TARGET_PROF_ENGAGED flag is set indicating target was engaged while flying a profile. AGPhase1 will skip any targets flagged as engaged from a profile.

Block 5. If weapon type is not Freefall, Programmed or Warhead, the result of the engagement is evaluated by the weapon rather that the launching AGAttacker.

Block 6. EngageControl is called to reset the AGAttacker to normal flight mode.

Block 7. If the AGAttacker is a flight leader and has launched on a commanded target with a weapon that is not Freefall, Programmed or Warhead, the engagement is ended for the AGAttacker.

Block 8. Otherwise, the target select phase for the AGAttacker is rescheduled at its user-specified start time.

Block 9. CheckAOIPattern is called to determine if an AOI search pattern should be adopted. A pattern will be adopted if the AGAttacker platform is not currently, orbiting a target and a target type of TARGET_AOI is found in the platforms target list that is within the user-specified range to initiate a search profile. A target type of TARGET_AOI represents the centroid of a target area or strike package.

Block 10. If the AGAttacker is a flight leader and the target being processed is a target of opportunity and the launched weapon is not an ARM, the complete message is broadcast.

Block 11. If BDA has been performed, the message flag for the engagement complete message sent to the flight leader is set indicating commanded BDA has been performed.

Block 12. EngageSuccess is called to determine the result of the engagement (i.e. Kill/No kill). The status returned can be one of the following: Failed Dead Target, Failed Target Moved, Hitbase, or Success.

Block 13. EngageControl is called to record the engagement status returned from EngageSuccess.

Block 14. EngageControl is called to record the engagement status returned from EngageSuccess for successful engagement as KILL_PROXIMITY or KILL_INTERCEPT.

Block 15. BattleDamageAssessment is called to perform battle damage assessment on a particular target platform. The outcome of the assessment is based on user-input probabilities of assessment as a function of track quality (confidence level) and target truth state.

Block 16. If an area kill assessment is being performed, all platforms in the scenario are checked to determine is they were within the lethal radius of the weapon intercept.

Block 17. Loops over all OPFACS in the OPFAC list.

Block 18. EngageControl is called to record the engagement status returned from EngageSuccess for successful engagement as KILL_PROXIMITY.

AGPhase7

The C3I module, AGPhase7, is the module that processes the react to engagement phase for the AGAttacker ruleset. Inputs and outputs for AGPhase7 are listed in Table 2.11-15 and the steps performed by the module are shown in Figure 2.11-30.

TABLE 2.11-15. Inputs and Outputs for AGPhase7.

Inputs:	
SimTime	Current Simulation Time
Order	Pointer to Engagement Order Struct
Outputs:	
Return Status	Returns 0 (zero) for success

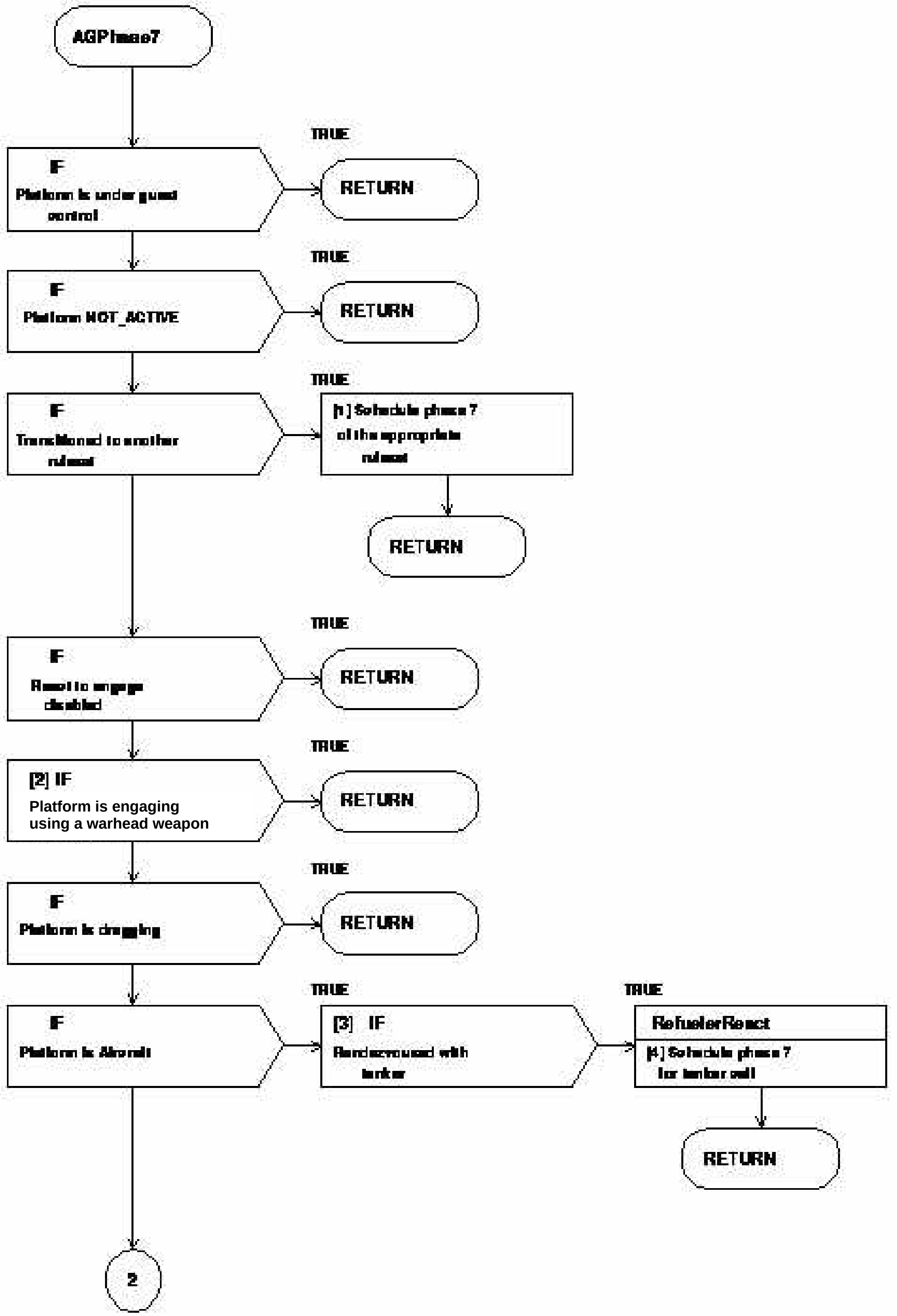

FIGURE 2.11-30. Flow Diagram for AGPhase 7: React to Engage.

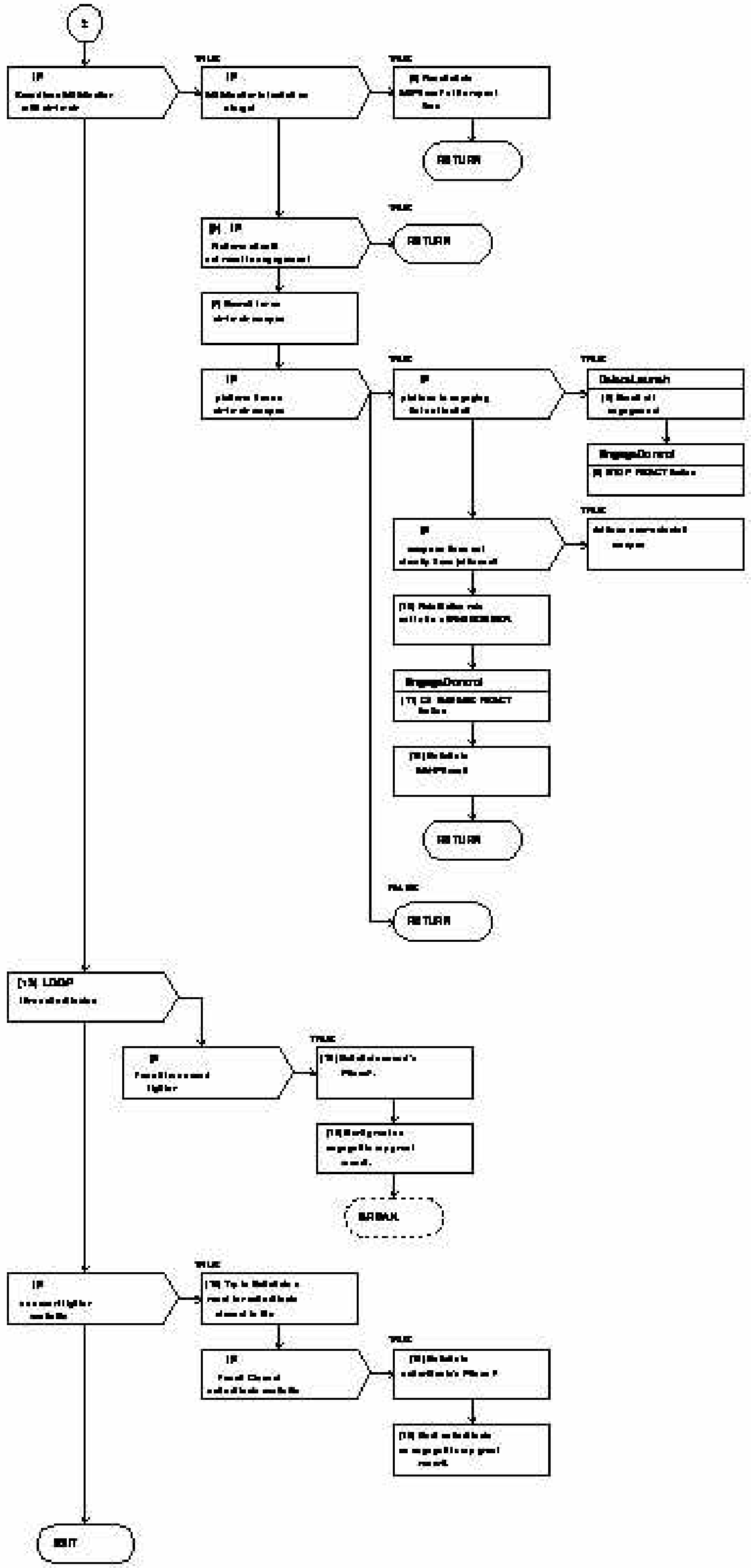

FIGURE 2.11-30. Flow Diagram for AGPhase 7: React to Engage. (Contd.)

Block 1. If the platform has transitioned another ruleset, schedule the reaction phase for the new ruleset immediately.

Block 2. If the platform is engaging with a warhead weapon, it is a missile platform; missiles do not react to engagement, therefore, exit the phase.

Block 3. If air refueling has begun, schedule reaction for tanker cell.

Block 4. If AGAttacker is currently locked on a target, do not react immediately, reschedule the reaction phase at its repeat time.

Block 5. A check is made to determine if platform should react. This is based on a random draw and compared to user-specified probabilities.

Block 6. Loop through weapons searching for an air-to-air weapon that could be used against the threat.

Block 7. Since we have not locked on target, delete the launch record in preparation for defensive action.

Block 8. EngageControl is called for the STOP_REACT action. This represents the stop action against the old target. A stop action message is sent to flight processing.

Block 9. The AGAttacker transitions to the WAS BOMBER ruleset in preparation to defend itself against the threat.

Block 10. EngageControl is called to record the ENGAGE_REACT action. Flags are set indicating engagement of the threat. An engage message is sent to flight processing.

Block 11. Since the AGAttacker has transitioned to the WAS BOMBER ruleset, the WAS BOMBER engagement phase (AAHPhase4) is scheduled at its start time.

Block 12. The AGAttacker has escorts, loop over all subordinates to find an escort that can engage the threat, i.e. is not currently engaging another target.

Block 13. If an escort is found, schedule the reaction phase for the escort fighter to engage the threat immediately.

Block 14. Set flags for the escort as engager of the threat.

Block 15. If an unengaged escort is not found, select the subordinate closest to the AGAttacker and schedule the subordinates reaction phase.

AGPhase8

The C3I module, AGPhase8, is the module that processes the react to lock phase for the AGAttacker ruleset. Inputs and outputs for AGPhase8 are listed in Table 2.11-16 and the steps performed by the module are shown in Figure 2.11-31.

TABLE 2.11-16. Inputs and Outputs for AGPhase8.

Inputs:	
SimTime	Current Simulation Time
Order	Pointer to Engagement Order Struct
Outputs:	
Return Status	Returns 0 (zero) for success

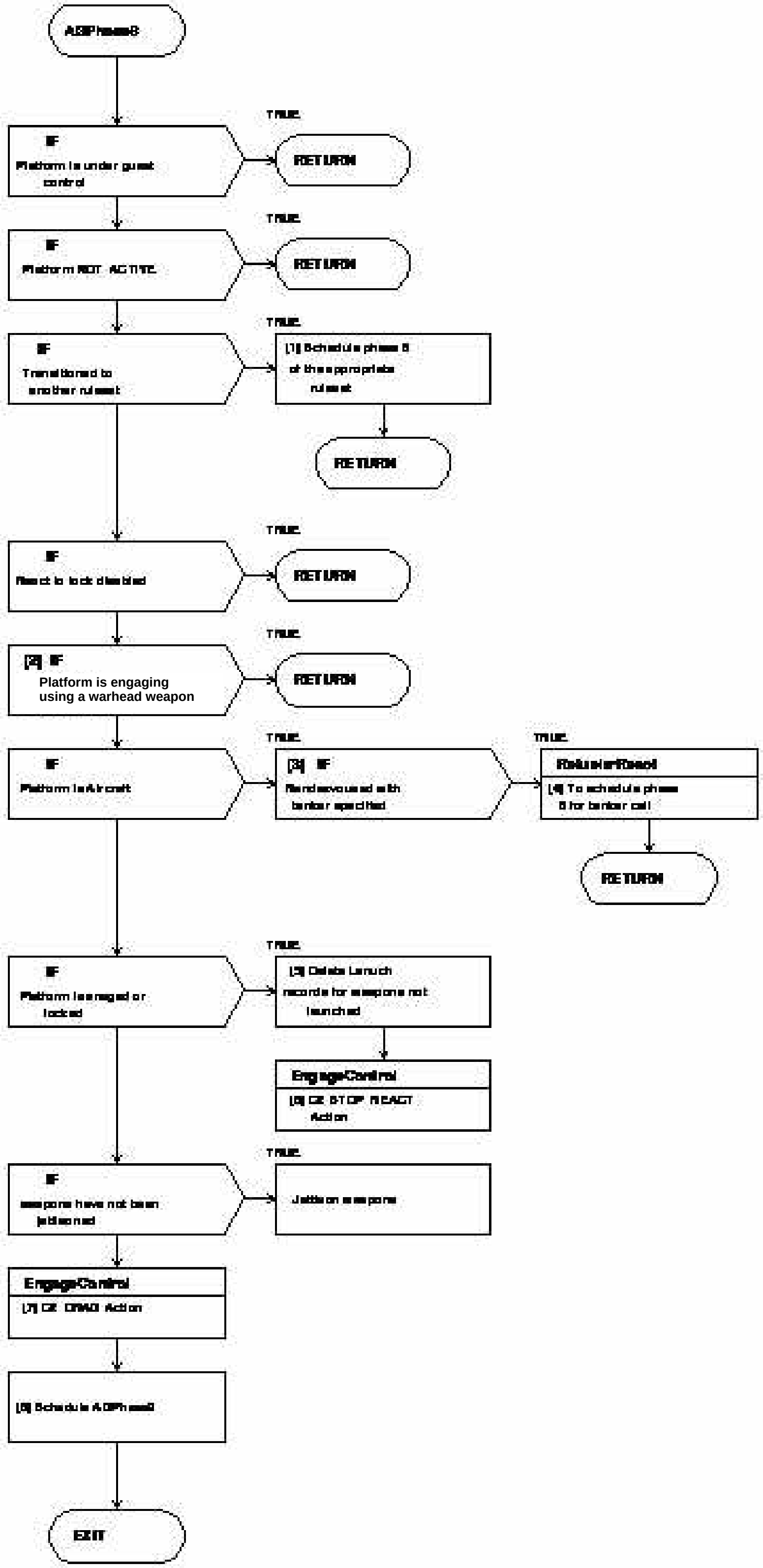
AGPhase8
TRUE
IF
Platform is under guest control
RETURN
TRUE
IF
Platform NOT ACTIVE
RETURN
TRUE
IF
Transitioned to another ruleset
[1] Schedule phase 8 of the appropriate ruleset
RETURN
TRUE
IF
React to lock disabled
RETURN
TRUE
[2] IF
Platform is engaging using a warhead weapon
RETURN
TRUE
IF
Platform is Aircraft
TRUE
[3] IF
Rendezvoused with tanker specified
TRUE
RefuelerReact
[4] To schedule phase 8 for tanker cell
RETURN
TRUE
IF
Platform is engaged or locked
[5] Delete Lanuch records for weapons not launched
EngageControl
[6] CE STOP REACT Action
TRUE
IF
weapons have not been jettisoned
Jettison weapons
EngageControl
[7] CE DRAG Action
[8] Schedule AGPhase8
EXIT

FIGURE 2.11-31. Flow Diagram for AGPhase 8: React to Lock.

Block 1. If the platform has transitioned another ruleset, schedule the react to lock phase for the new ruleset immediately.

Block 2. If the platform is engaging with a warhead weapon, it is a missile platform; missiles do not react to lock, therefore, exit the phase.

Block 3. Check to see if aircraft platform has rendezvoused with the tanker.

Block 4. If air refueling has begun, schedule react to lock for tanker cell.

Block 5. Delete all the launch records for any weapons that have not been launched in preparation for defensive action.

Block 6. EngageControl is called for the STOP_REACT action. This represents the stop action against the old target. A stop message is sent to flight processing.

Block 7. EngageControl is called to record the DRAG action. Flags are set indicating initiation of drag maneuver.

Block 8. AGPhase9 (i.e. AGAttacker Completion of Drag Phase) is scheduled at its user-specified start time.

AGPhase9

The C3I module, AGPhase9, is the module that processes the completion of drag phase for the AGAttacker ruleset. Inputs and outputs for AGPhase9 are listed in Table 2.11-17 and the steps performed by the module are shown in Figure 2.11-32.

TABLE 2.11-17. Inputs and Outputs for AGPhase9.

Inputs:	
SimTime	Current Simulation Time
Order	Pointer to Engagement Order Struct
Outputs:	
Return Status	Returns 0 (zero) for success

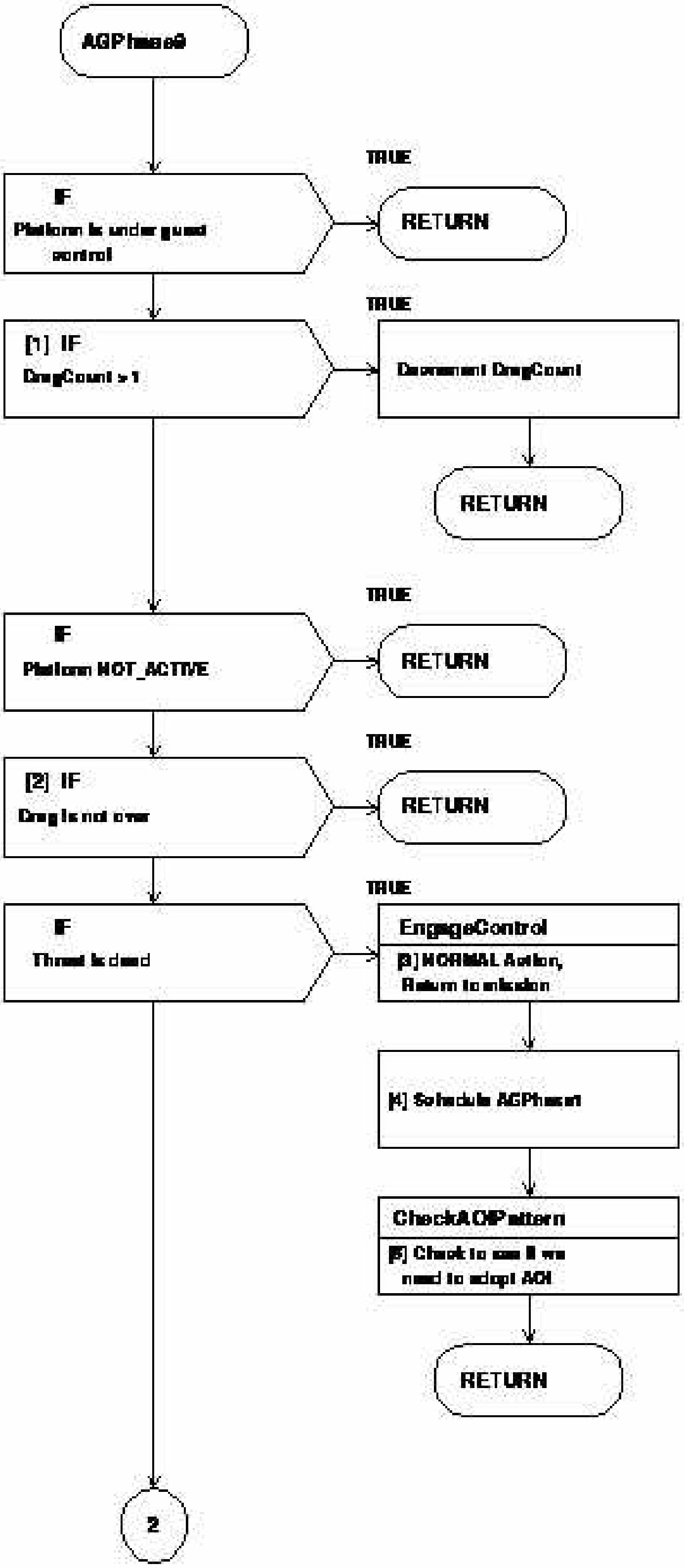

FIGURE 2.11-32. Flow Diagram for AGPhase9: Drag.

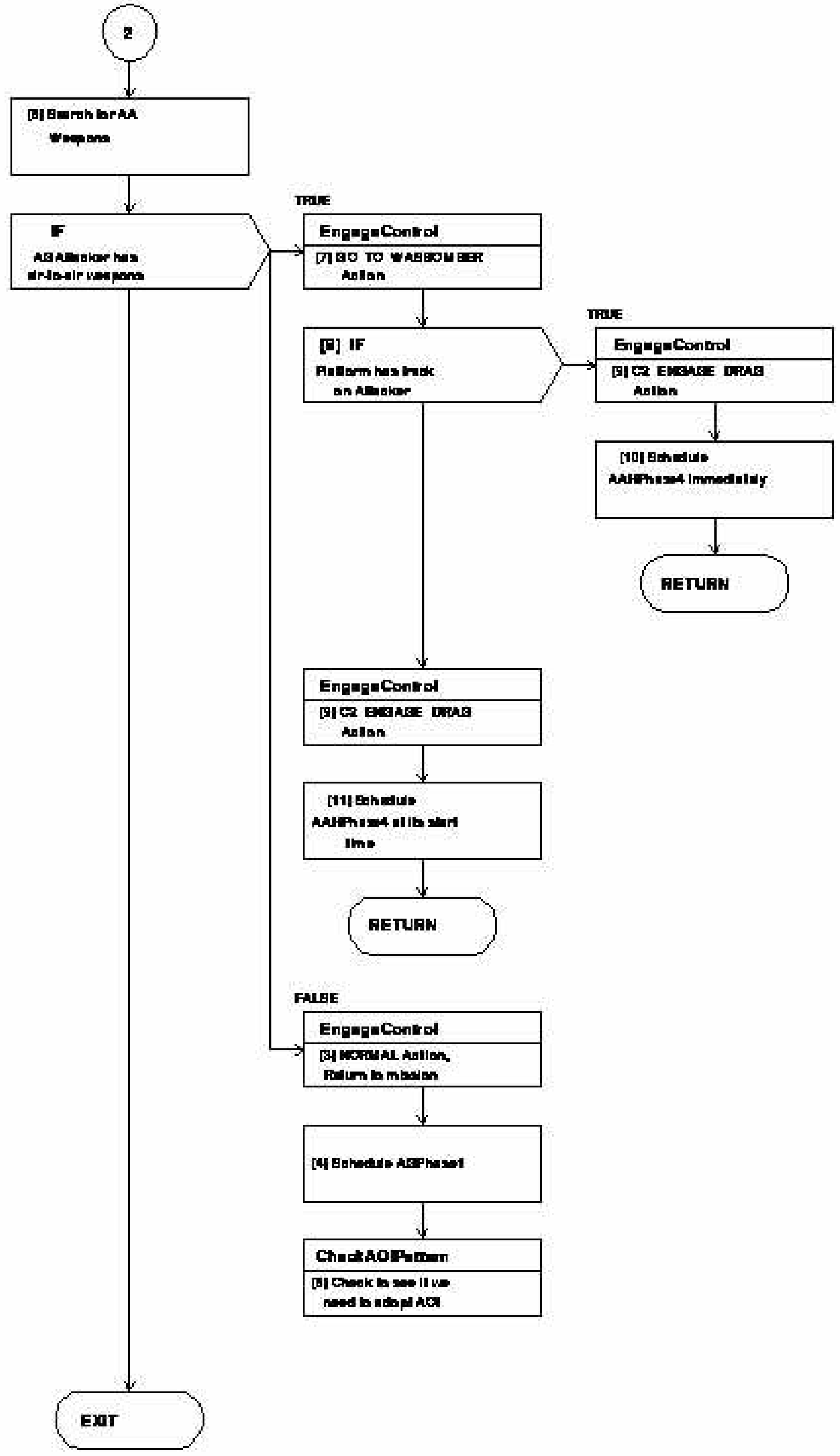
2
[6] Search for AA Weapons
IF
AG Attacker has air-to-air weapons
TRUE
EngageControl
[7] GO TO WASBOMBER Action
[8] IF
Platform has track on Attacker
TRUE
EngageControl
[9] C2 ENGAGE DRAG Action
[10] Schedule AAHPhase4 Immediately
RETURN
EngageControl
[9] C2 ENGAGE DRAG Action
[11] Schedule AAHPhase4 at its start time
RETURN
FALSE
EngageControl
[3] NORMAL Action, Return to mission
[4] Schedule AGPhase1
CheckAOIPattern
[5] Check to see if we need to adopt AOI
EXIT

FIGURE 2.11-32. Flow Diagram for AGPhase9: Drag. (Contd.)

Block 1. A check is made to determine if more than one drag maneuver is being performed. Drag maneuvers can be queued in response to engagement by multiple SAM sites.

Block 2. A check is made to determine if any platform has missiles in the air against the AGAttacker. If so, continue dragging.

Block 3. EngageControl is called to record the NORMAL action. This indicates that normal flight mode is being adopted and platform is returning to its mission.

Block 4. Upon return to normal flight, the target select phase for the AGAttacker is scheduled at its start time.

Block 5. CheckAOIPattern is called to determine if an AOI search pattern should be adopted. A pattern will be adopted if the AGAttacker platform is not currently, orbiting a target and a target type of TARGET_AOI is found in the platforms target list that is within the user-specified range to initiate a search profile. A target type of TARGET_AOI represents the centroid of a target area or strike package.

Block 6. Loop through weapons searching for an air-to-air weapon that could be used against the threat.

Block 7. EngageControl is called to transition AGAttacker to the WAS BOMBER ruleset in preparation to defend itself against the threat.

Block 8. A check is made to determine if AGAttacker has track information on threat.

Block 9. EngageControl is called to record the ENGAGE_DRAG action. Flags are set indicating platform as engager on completion of drag maneuver.

Block 10. The AGAttacker has transitioned to the WAS BOMBER ruleset. The WAS BOMBER engagement phase (AAHPhase4) is scheduled immediately because we have track information on threat.

Block 11. The AGAttacker has transitioned to the WAS BOMBER ruleset. The WAS BOMBER engagement phase (AAHPhase4) is scheduled at its start time because we do not yet have track information on threat. This will allow more time to obtain track on threat.

AGPhase12

The C3I module, AGPhase12, is the module that processes the react to SAM lock phase for the AGAttacker ruleset. Inputs and outputs for AGPhase12 are listed in Table 2.11-18 and the steps performed by the module are shown in Figure 2.11-33.

TABLE 2.11-18. Inputs and Outputs for AGPhase12.

Inputs:	
SimTime	Current Simulation Time
Order	Pointer to Engagement Order Struct
Outputs:	
Return Status	Returns 0 (zero) for success

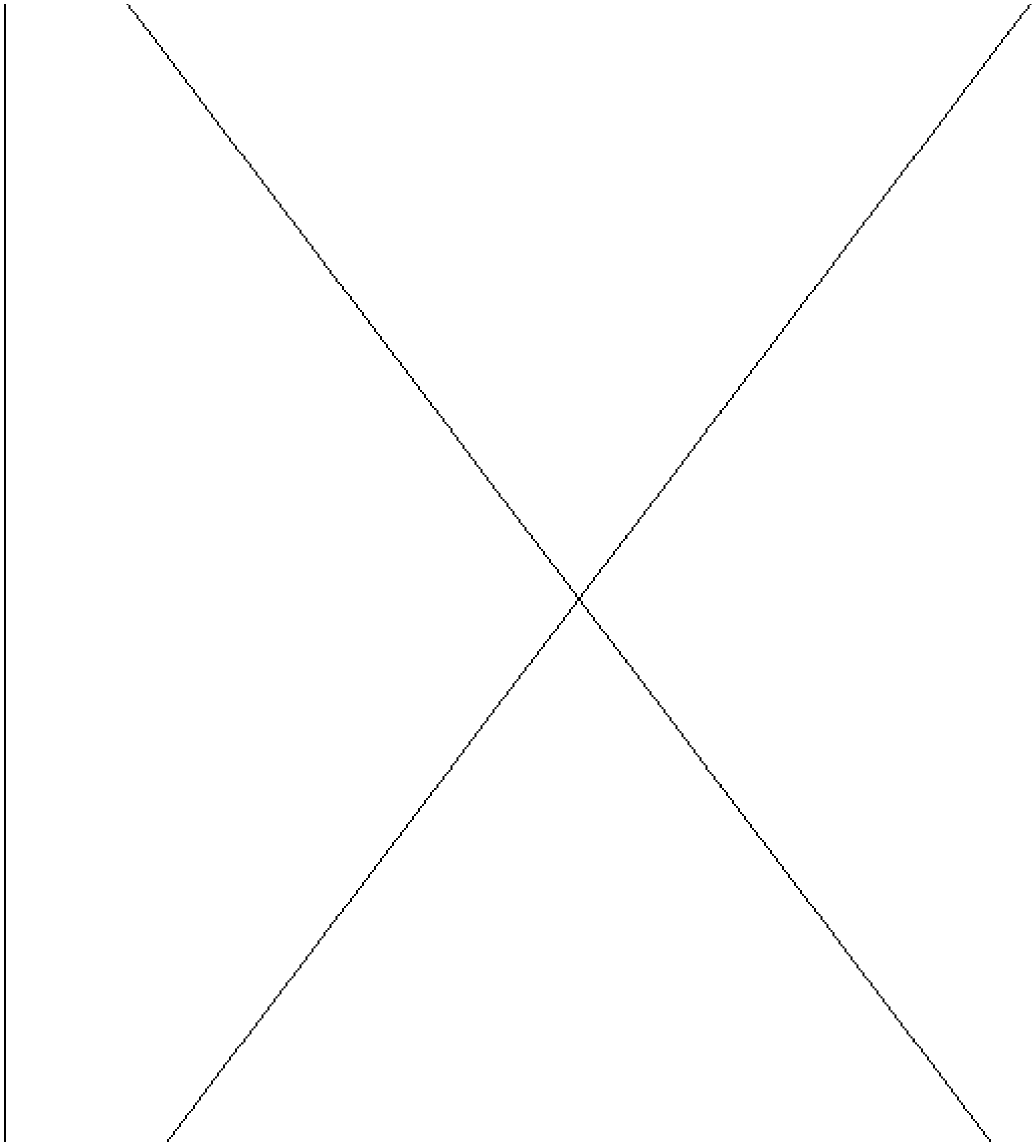

FIGURE 2.11-33. Flow Diagram for AGPhase12: React to SAM Lock.

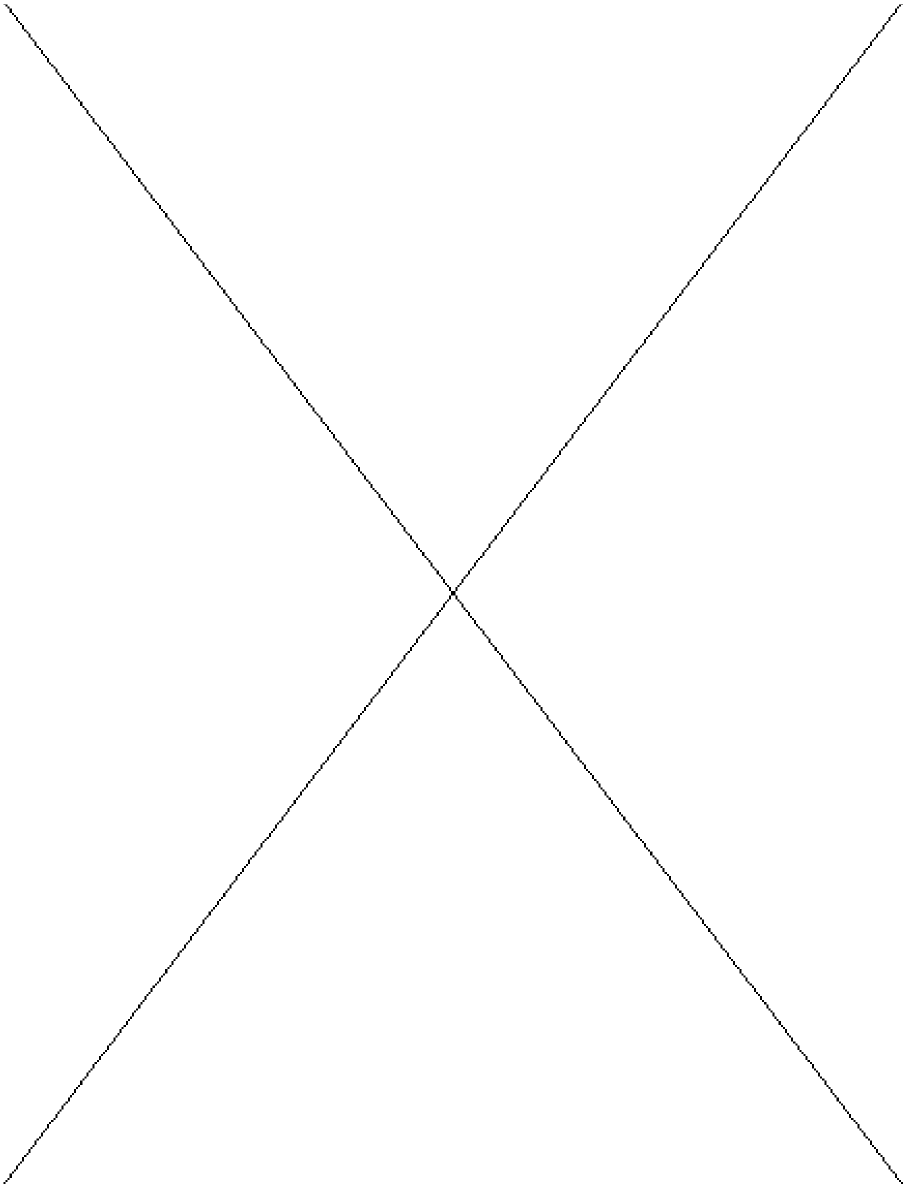

FIGURE 2.11-33. Flow Diagram for AGPhase12: React to SAM Lock. (Contd.)

<u>Block 1.</u> The condition is checked to determine if engaging with warhead weapon. This indicates that the AGAttacker is modeling a missile platform and missiles do not react to lock, therefore, exit the phase.

<u>Block 2.</u> If platform has already transition to WAS BOMBER, exit the phase. WAS BOMBER ruleset is handling engagement.

<u>Block 3.</u> If AGAttacker has engaged or locked but not launched on target, reschedule AGPhase12 at its repeat time. Engaging the target is higher priority than reacting to SAM lock.

<u>Block 4.</u> If avoid maneuver has not begun, begin maneuver now.

Block 5. EngageControl is called to initiate the avoid maneuver. The platform enters avoid mode where it will fly normal to the threat (this is referred to as the "first move").

Block 6. EngageControl is called to fly platform to the avoid altitude.

Block 7. Now that the avoid maneuver has been initiated, reschedule AGPhase12 at current simulation time + AvoidDownTime. The maneuver (i.e. first move) will be flown for the user specified avoidance time (i.e. AvoidDownTime).

Block 8. Check to see if "first move" has been completed.

Block 9. AvoidCount indicates avoiding for more than one SAM LOCK. If other avoid maneuvers need to be processed, decrement count and exit.

Block 10. A check is made to determine if the current active phase for AGAttacker is AGPhase12.

Block 11. EngageControl is called to transition back to NORMAL flight mode and return to mission.

Block 12. EngageControl is called to transition to return to base (RTB) flight mode. Platform/flight will return to home airbase.

Block 13. EngageControl is called to command platform to fly at the user-specified avoid altitude.

Block 14. The end avoid/end react is scheduled at current SimTime + ResumeDownTime. The platform will terminate avoid and return to flying waypoints. However, the platform will remain at the AvoidDownAlttitude for the ResumeDownTime.

Block 15. SendMeHome is called to schedule the platform to return to base.

Block 16. CheckAOIPattern is called to determine if an AOI search pattern should be adopted. A pattern will be adopted if the AGAttacker platform is not currently orbiting a target and a target type of TARGET_AOI is found in the platforms target list that is within the user-specified search profile initiation range of the AGAttacker platform. A target type of TARGET_AOI represents the centroid of a target area or strike package.

CheckPattern

The C3I module, CheckPattern, is the module that determines when search/track profiles should be initiated for the AGAttacker ruleset. Inputs and outputs for CheckPattern are listed in Table 2.11-19 and the steps performed by the module are shown in Figure 2.11-34.

TABLE 2.11-19. Inputs and Outputs for CheckPattern.

Inputs:	
SimTime	Current Simulation Time
OpFac	Pointer to OpFac Struct
TrackFile	Pointer to Track Header Struct
Outputs:	
Return Status	Returns 0 (zero) for success

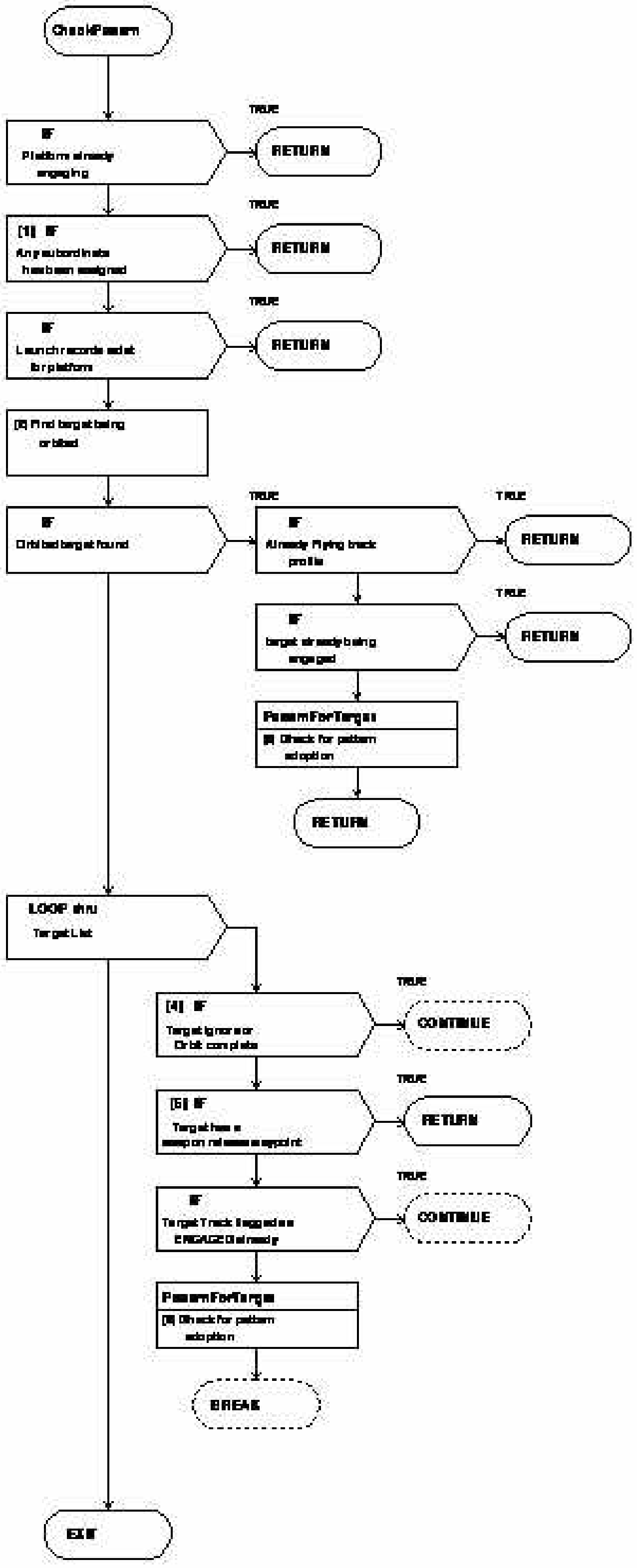
CheckPattern
TRUE
IF
Platform already engaging
RETURN
TRUE
[1] IF
Any subordinate has been assigned
RETURN
TRUE
IF
Launch records exist for platform
RETURN
[2] Find target being orbited
TRUE
IF
Orbited target found
TRUE
IF
Already Flying track profile
RETURN
TRUE
IF
target already being engaged
RETURN
PatternForTarget
[3] Check for pattern adoption
RETURN
LOOP thru
Target List
TRUE
[4] IF
Target ignore or Orbit complete
CONTINUE
TRUE
[5] IF
Target has a weapon release waypoint
RETURN
TRUE
IF
Target Track flagged as ENGAGED already
CONTINUE
PatternForTarget
[3] Check for pattern adoption
BREAK
EXIT

FIGURE 2.11-34. Flow Diagram for CheckPattern.

Block 1. If any AGAttacker subordinate has been assigned a target for engagement, exit this module. Profiles are not initiated if wingmen are engaging threats.

Block 2. All targets on target list are searched to find a target being orbited. This indicates that a profile is currently being flown for the orbited target.

Block 3. PatternForTarget is called to determine if the AGAttacker platform should initiate a search profile, continue search profile, or transition from search to track profile.

Block 4. A check is made to determine if the target is marked as IGNORE or the number of orbits specified on the profile definition have been completed.

Block 5. If the AGAttacker platform has a scripted target with a weapon release waypoint specified, patterns will not be adopted.

CheckARMPattern

The C3I module, CheckARMPattern, is the module used by AGAttacker to determine if a search profile should be updated for ARM platform flight. Inputs and outputs for CheckARMPattern are listed in Table 2.11-20 and the steps performed by the module are shown in Figure 2.11-35.

TABLE 2.11-20. Inputs and Outputs for CheckARMPattern.

Inputs:	
SimTime	Current Simulation Time
OpFac	Pointer to OpFac Struct
Outputs:	
Return Status	Returns 0 (zero) for success

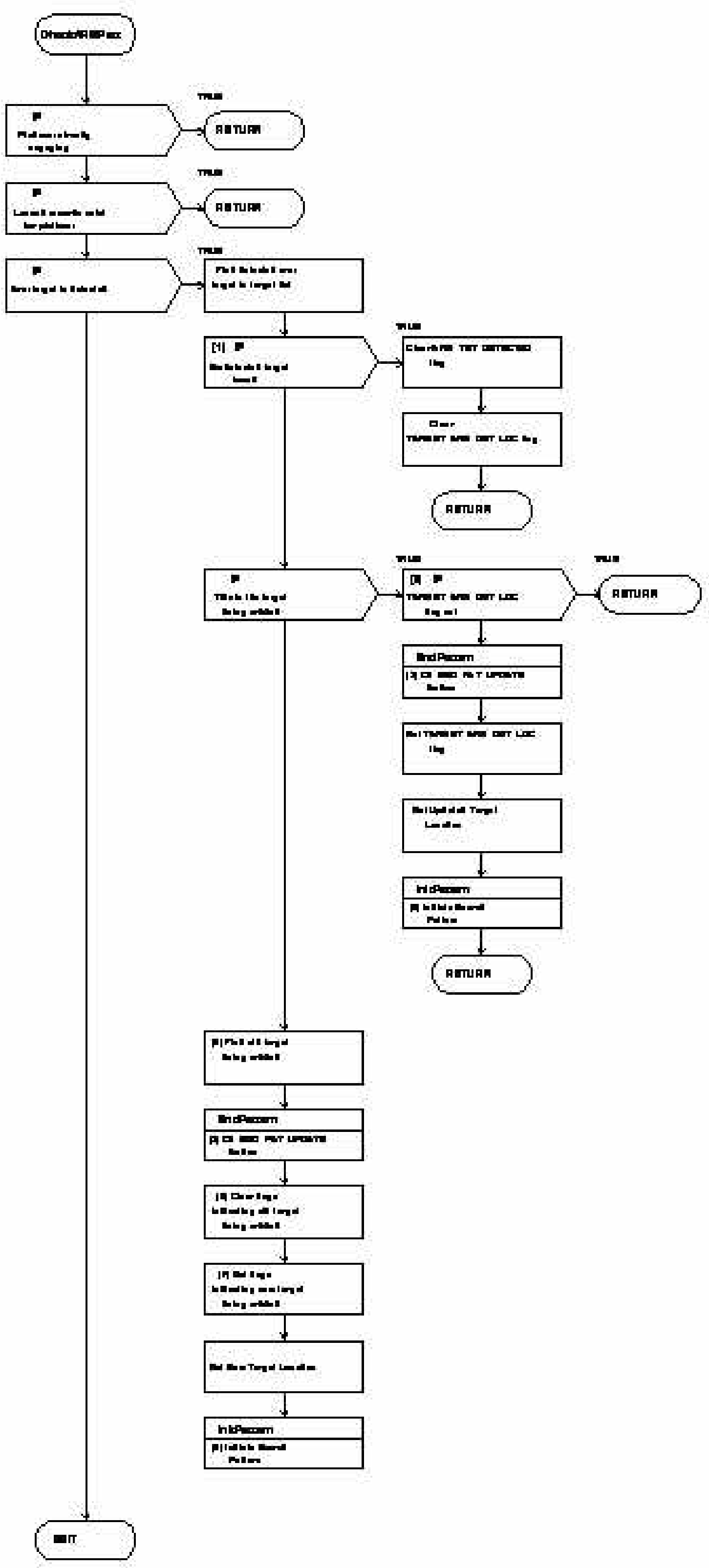

FIGURE 2.11-35. Flow Diagram for CheckARMPattern.

Block 1. If no target was found in the target list marked as ARM_TGT_DETECTED, the flags are cleared indicating loss of track on target. The module is exited to allow target select phase to continue being scheduled.

Block 2. If TARGET_ARM_DET_LOC is set, this indicates that the profile has already been rescaled for the detected target location, nothing else to do.

Block 3. EndPattern is called for the C2_END_PAT_UPDATE indicating that the current profile is being ended and a new rescaled profile is being initiated.

Block 4. InitPattern is called to rescale and initiate the profile based on the detected target location.

Block 5. The target found marked as ARM_TGT_DETECTED was not the target currently being orbited. Search the target list to find the old target being orbited.

Block 6. Clear flags on old target indicating that old target no longer being orbited.

Block 7. Set flags on new target indicating that new target is being orbited.

SelectAttackWeapon

The C3I module, SelectAttackWeapon, is the module used by AGAttacker to select the best weapon for engagement. Inputs and outputs for SelectAttackWeapon are listed in Table 2.11-21 and the steps performed by the module are shown in Figure 2.11-36.

TABLE 2.11-21. Inputs and Outputs for SelectAttackWeapon.

Inputs:	
SimTime	Current Simulation Time
OpFac	Pointer to OpFac Struct
Target	Pointer to Target Struct
UseScripted	UseScripted Weapon Indicator
CanAssign	Can Assign Subordinate Indicator
Outputs:	
WpnCount	Pointer to WpnCount Variable
Status	Pointer to Status Variable
PK	Pointer to PK Variable
Return Weapon	Returns Pointer to WeaponStruct

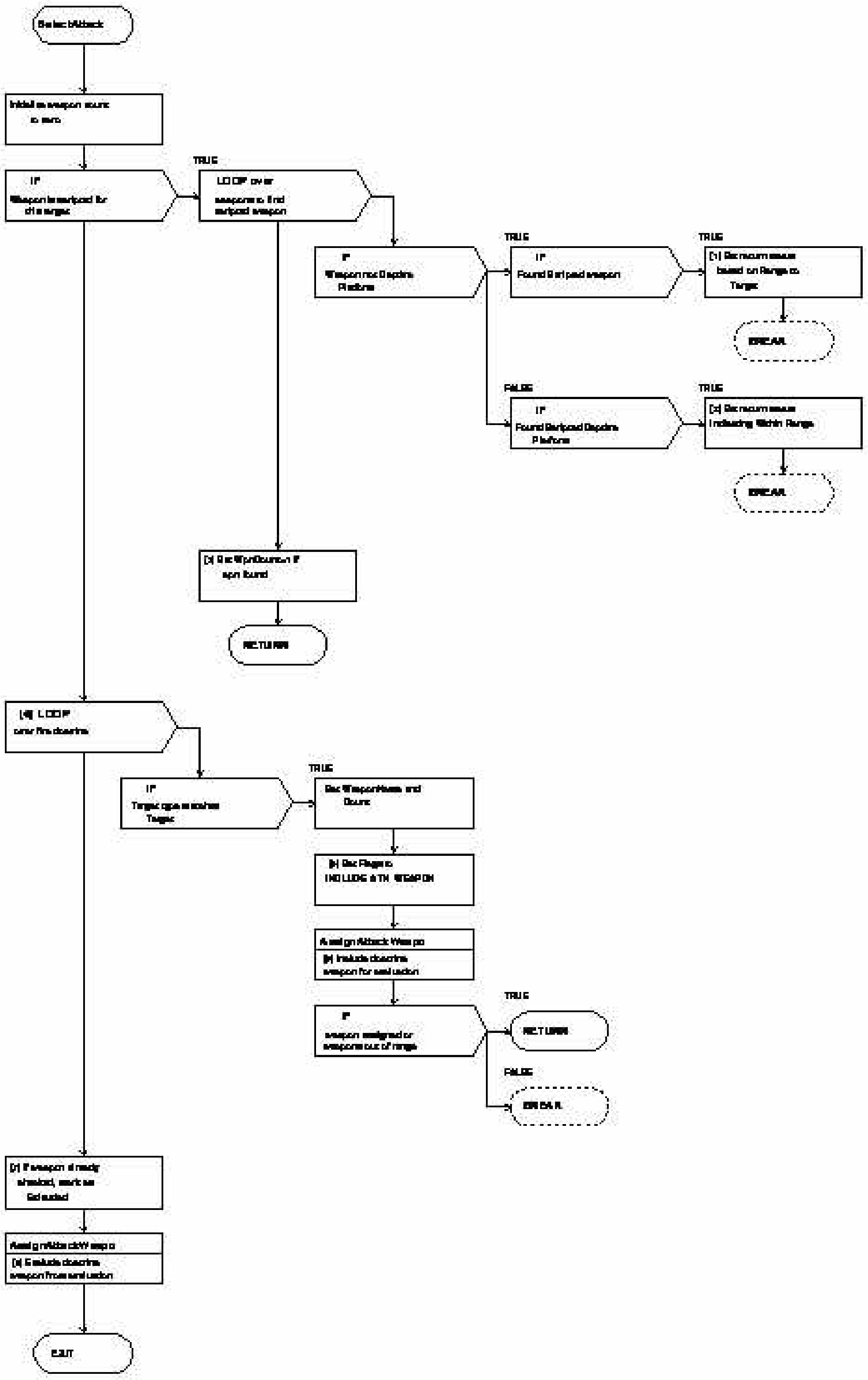
SelectAttack
Initialize weapon count to zero
IF Weapon is assigned for this target
TRUE
LOOP over weapons to find assigned weapon
IF Weapon not Depletes Platform
TRUE
IF Found assigned weapon
TRUE
[1] Set return values based on Range to Target
BREAK
FALSE
IF Found Assigned Depletes Platform
TRUE
[2] Set return values Indicating Within Range
BREAK
[3] Set WpnCount to 1 if wpn found
RETURN
[4] LOOP over fire doctrine
IF Target type matches Target
TRUE
Set WeaponName and Count
[5] Set Flag to INCLUDE ATK WEAPON
AssignAttackWeapo
[6] Include doctrine weapon for evaluation
IF weapon assigned or weapons out of range
TRUE
RETURN
FALSE
BREAK
[7] If weapon already assigned, mark as Excluded
AssignAttackWeapo
[8] Exclude doctrine weapon from evaluation
EXIT

FIGURE 2.11-36. Flow Diagram for SelectAttackWeapon.

Block 1. The return Status variable is set based on the Range to target. The Status value indicates either platform is within weapon max range or outside weapon max range.

Block 2. The return Status variable is set indicating that the platform is within the captive deployment range. The captive deployment range check was made in AGPhase1 prior to calling SelectAttackWeapon.

Block 3. The return value for WpnCount is set to 1 if a weapon was selected.

Block 4. If no weapon is scripted for target and fire doctrine has been specified in the AGAttacker ruleset, the target type/weapon combinations in the fire doctrine are used to select a weapon.

Block 5. If the doctrine target type matches the target being checked for weapon selection, a flag is set to include the doctrine weapon type in the weapon selection process. For this “include” case, only the doctrine weapon is evaluated.

Block 6. AssignAttackWeapon is called to select a weapon for engagement including the doctrine weapon type.

Block 7. If the doctrine weapon type has already been check for selection, set a flag indicating that weapon selection should be tried again excluding the doctrine weapon.

Block 8. AssignAttackWeapon is called to select a weapon for engagement excluding the doctrine weapon type. For this “exclude” case, all weapons other than the doctrine weapon will be evaluated.

AssessAttackTargets

The C3I module, AssessAttackTargets, is the module used by AGAttacker to assess, update and prioritize the platform target list. Inputs and outputs for AssessAttackTargets are listed in Table 2.11-22 and the steps performed by the module are shown in Figure 2.11-37.

TABLE 2.11-22. Inputs and Outputs for AssessAttackTargets.

Inputs:	
SimTime	Current Simulation Time
OpFac	Pointer to OpFac Struct
Outputs:	
Return Status	Returns 0 (zero) for success

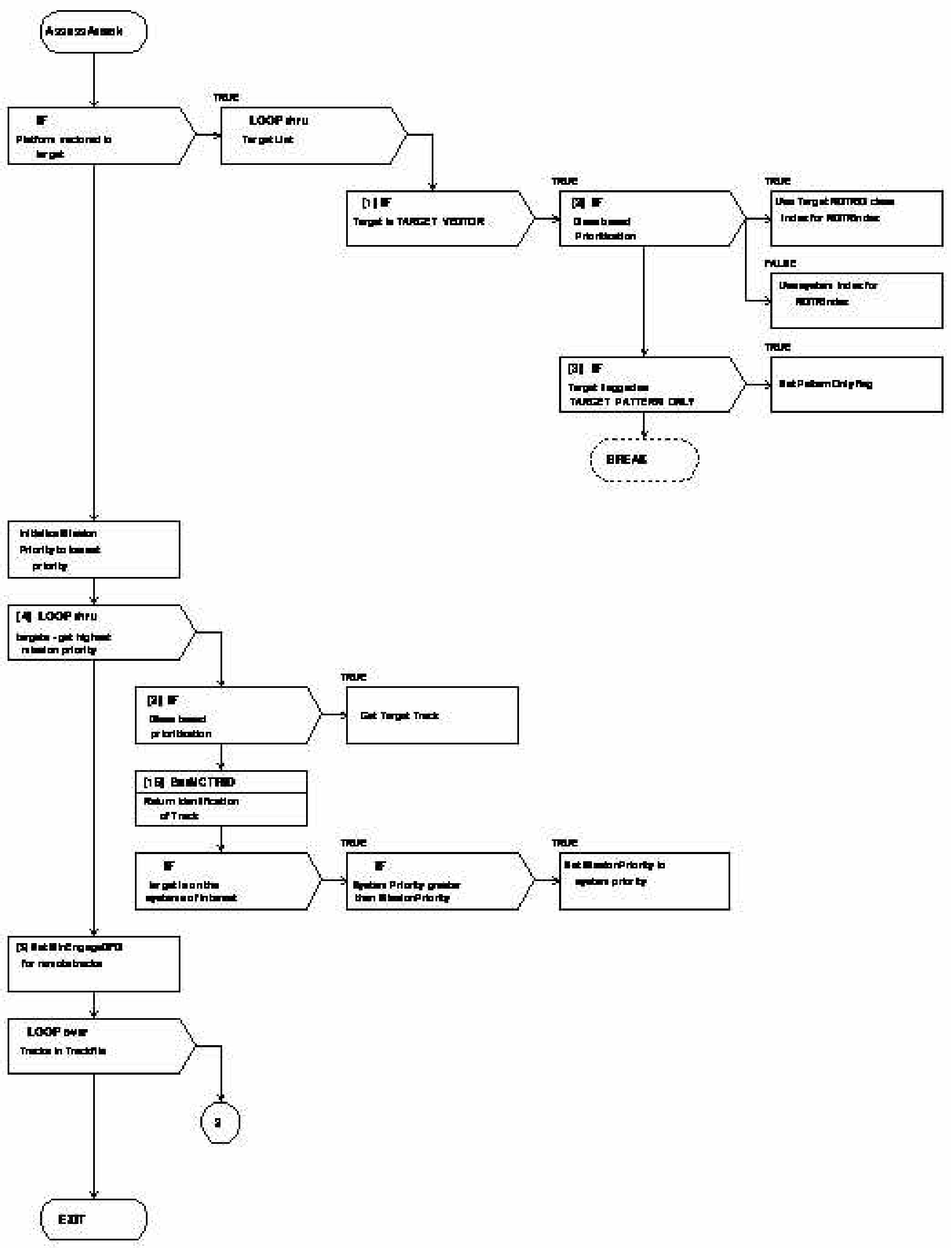
AssessAttack
IF
Platform vectored to target
TRUE
LOOP thru
Target List
TRUE
[1] IF
Target is TARGET VECTOR
TRUE
[2] IF
Class based Prioritization
TRUE
Use Target NOTRGI class Index for NOTRGIndex
FALSE
Use system Index for NOTRGIndex
TRUE
[3] IF
Target tagged as TARGET PATTERN ONLY
Set Pattern Only flag
BREAK
Initialize Mission Priority to lowest priority
[4] LOOP thru
targets - get highest mission priority
TRUE
[2] IF
Class based prioritization
Get Target Track
[15] BasicTRID
Return Identification of Track
TRUE
IF
target is on the systems of interest
TRUE
IF
System Priority greater than MissionPriority
Set MissionPriority to system priority
[5] Set MinEngageDFG for non-obs tracks
LOOP over
Tracks in Trackfile
2
EXIT

FIGURE 2.11-37. Flow Diagram for AssessAttackTargets.

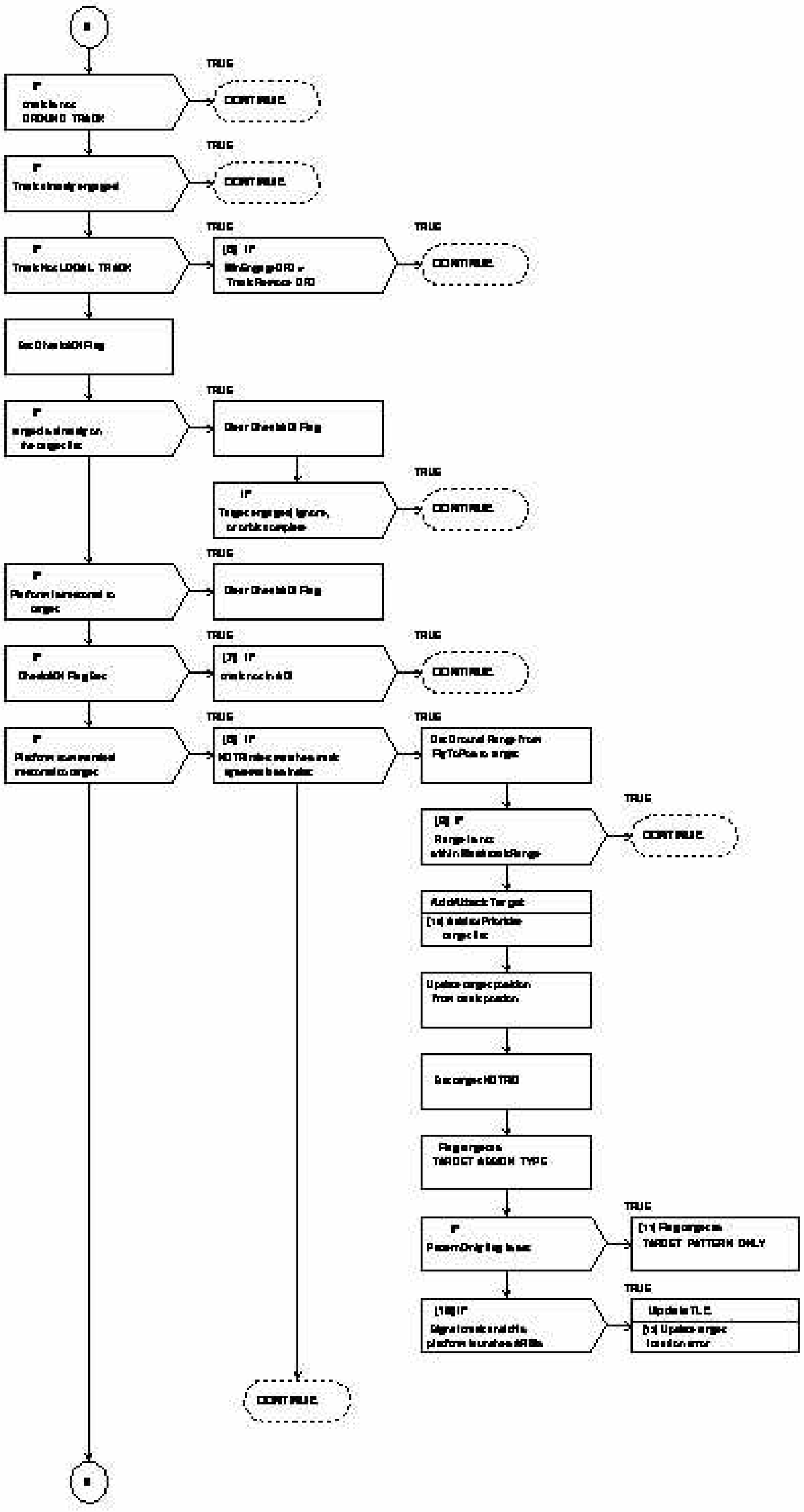

FIGURE 2.11-37. Flow Diagram for AssessAttackTargets. (Contd.)

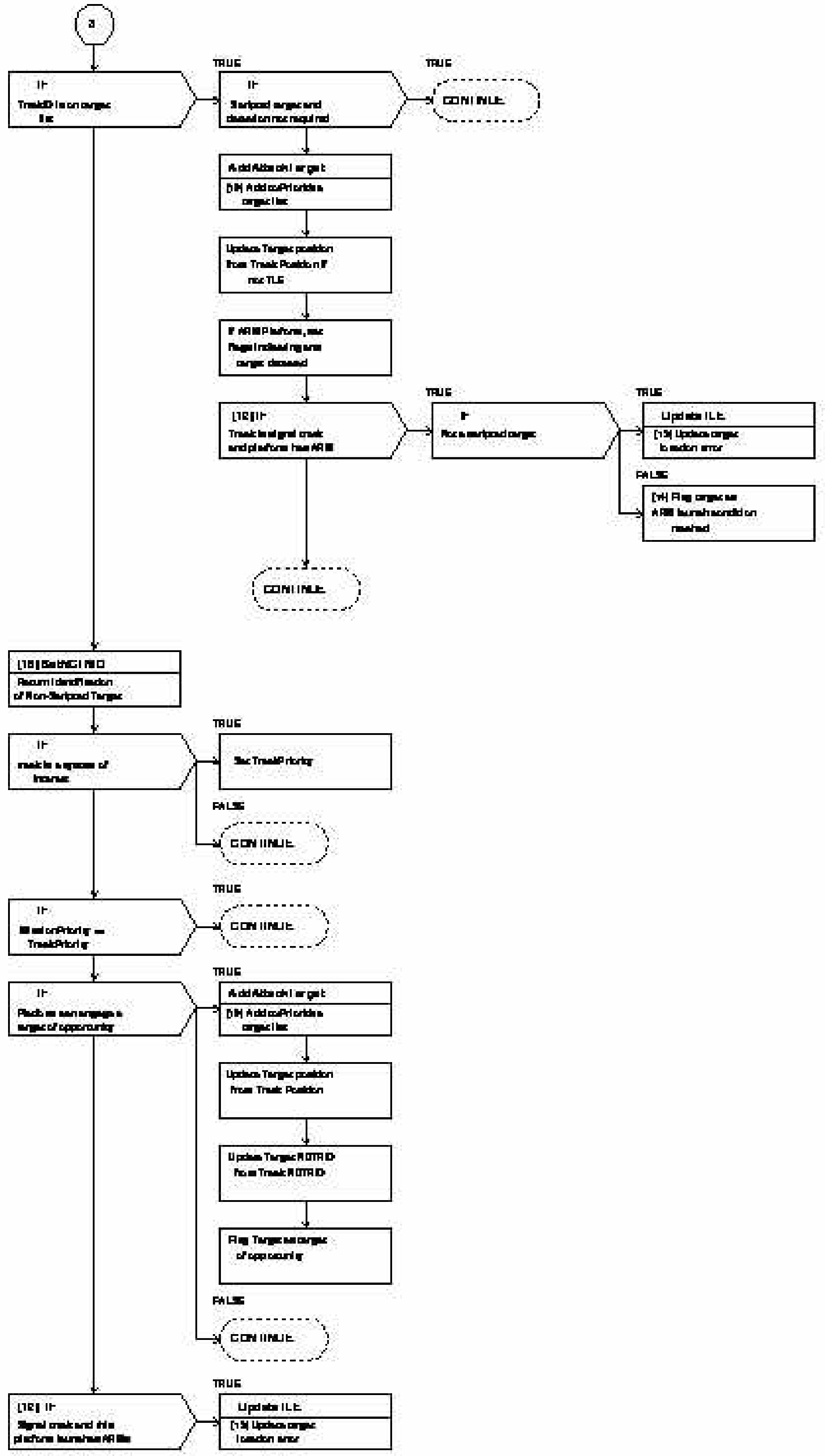
3
IF TrackID is on target list
TRUE
IF Stripped target and classification not required
TRUE
CONTINUE
AddAttackTarget
[11] Add to Prioritized target list
Update Target position from Track Position if not TLE
If ARM Platform, set flags indicating the target detected
TRUE
[12] IF Track is a signal track and platform has ARM
IF Not a stripped target
TRUE
Update TLE
[13] Update target location error
FALSE
[14] Flag target as ARM launch condition reached
CONTINUE
[15] SetNCTRID
Return Identification of Non-Stripped Target
IF track is a system of interest
TRUE
Set TrackPriority
FALSE
CONTINUE
IF MissionPriority == TrackPriority
TRUE
CONTINUE
IF Platform can engage a target of opportunity
TRUE
AddAttackTarget
[11] Add to Prioritized target list
Update Target position from Track Position
Update Target NCTRID from Track NCTRID
Flag Target as target of opportunity
FALSE
CONTINUE
[12] IF Signal track and this platform launches ARMs
TRUE
Update TLE
[13] Update target location error

FIGURE 2.11-37. Flow Diagram for AssessAttackTargets. (Contd.)

Block 1. A check is made to determine if this is the target that the platform has been vectored to.

Block 2. A check is made to determine if target prioritization is class based or system based. A class is a collection of system types. The NCTRIndex is set to the class index or system type index based on the check. NCTRIndex is used to compare other tracks to see if they fall in the same class or system type.

Block 3. A check is made to determine if target is flagged as PATTERN_ONLY. This indicates that the AGAttacker has no weapons and will pattern for targets but not select weapons and engage.

Block 4. Loop over the targets on the target list for the AGAttacker platform. The purpose of this loop is to determine the highest priority among the scripted targets.

Block 5. Search all weapons to find the lowest MinEngageDFD. MinEngageDFD is a value defining to lowest Data Functional Descriptor (DFD) for engaging a target specified by a particular track. Each sensor has a user-specified DFD rating (or accuracy/confidence level for track data).

Block 6. If MinEngageDFD is greater than the DFD rating of the track being assessed, continue to the next track. Tracks must meet MinEngageDFD for engagement to take place.

Block 7. If CheckAOI flag is set, determine if the track falls inside the user-specified AOI boundaries.

Block 8. A check is made to determine if track being assessed is of the same class or system as the target currently vectored to. If not, continue to next track.

Block 9. A check is made to determine if the ground range from flytopos (i.e. to vectored to position) to the target is within the AGAttacker MaxAttackRange. If not, continue to next track.

Block 10. AddAttackTarget is called to add tracks to the target list and prioritize the target list.

Block 11. The AGAttacker is currently vectored to a commanded target. If the target vectored to is a PATTERN_ONLY target, all assessed tracks matching its class/system type are marked as PATTERN_ONLY.

Block 12. A check is made to determine if the track is a signal track and if the AGAttacker has ARM weapons.

Block 13. UpdateTLE is called to update the errored target location and determines if arm launch criteria has been reached.

Block 14. Set target flags indicating that ARM weapon launch condition has been reached.

Block 15. SetNCTRID is called to return the identification of the track/target. The identification is either the name of a class or system type.

ProcessCmdTarget

The C3I module, ProcessCmdTarget, is the module used by AGAttacker to assess, update and prioritize the platform target list. Inputs and outputs for ProcessCmdTarget are listed in Table 2.11-23 and the steps performed by the module are shown in Figure 2.11-38.

TABLE 2.11-23. Inputs and Outputs for ProcessCmdTarget.

Inputs:	
SimTime	Current Simulation Time
OpFac	Pointer to OpFac Struct
Target	Pointer to Target Struct
Outputs:	
Return Status	0 - Exit AGPhase1 1 - Evaluate Weapon for Target 2 - Continue to next Target 3 - Reschedule AGPhase1

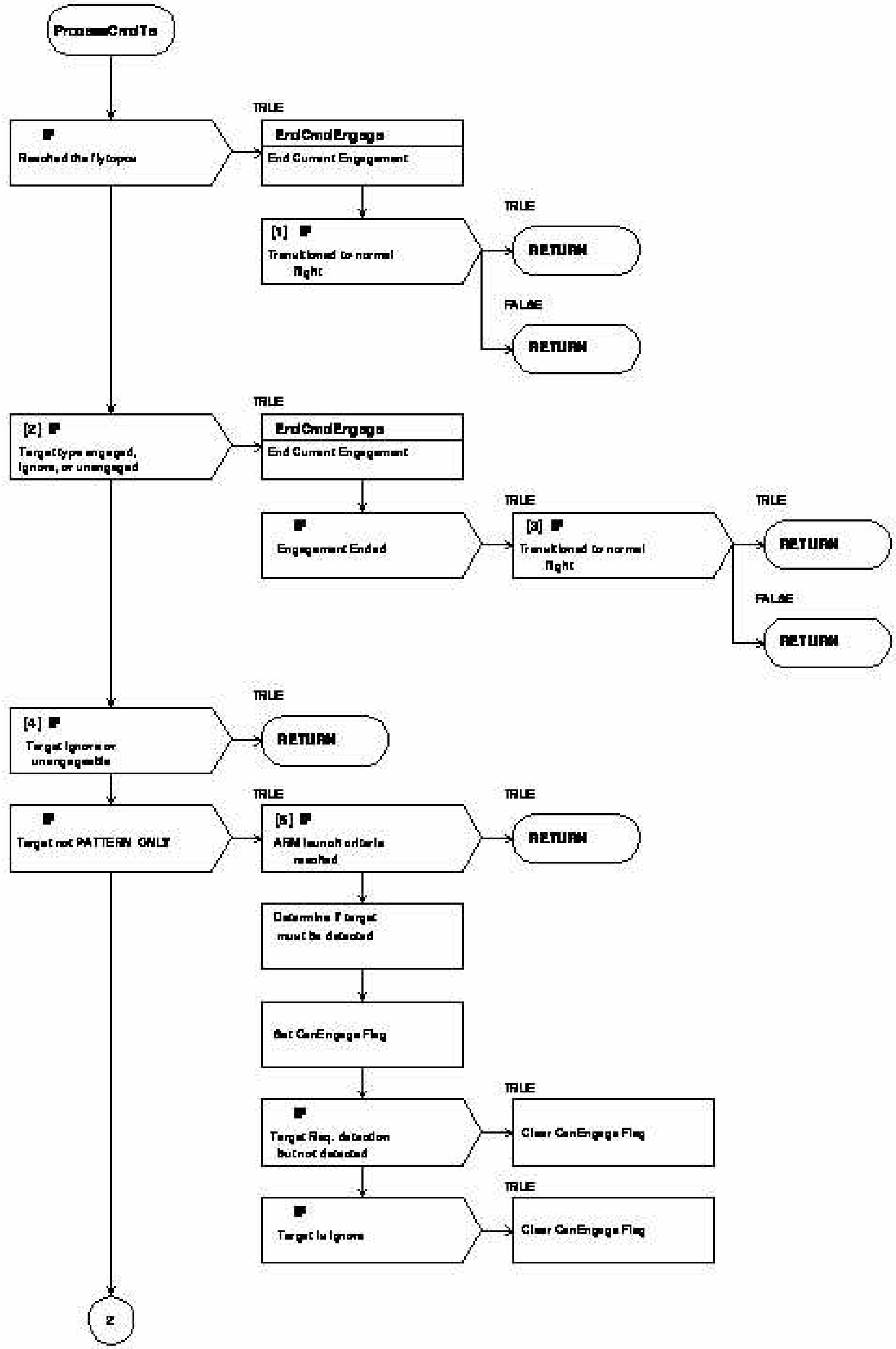

FIGURE 2.11-38. Flow Diagram for ProcessCmdTarget.

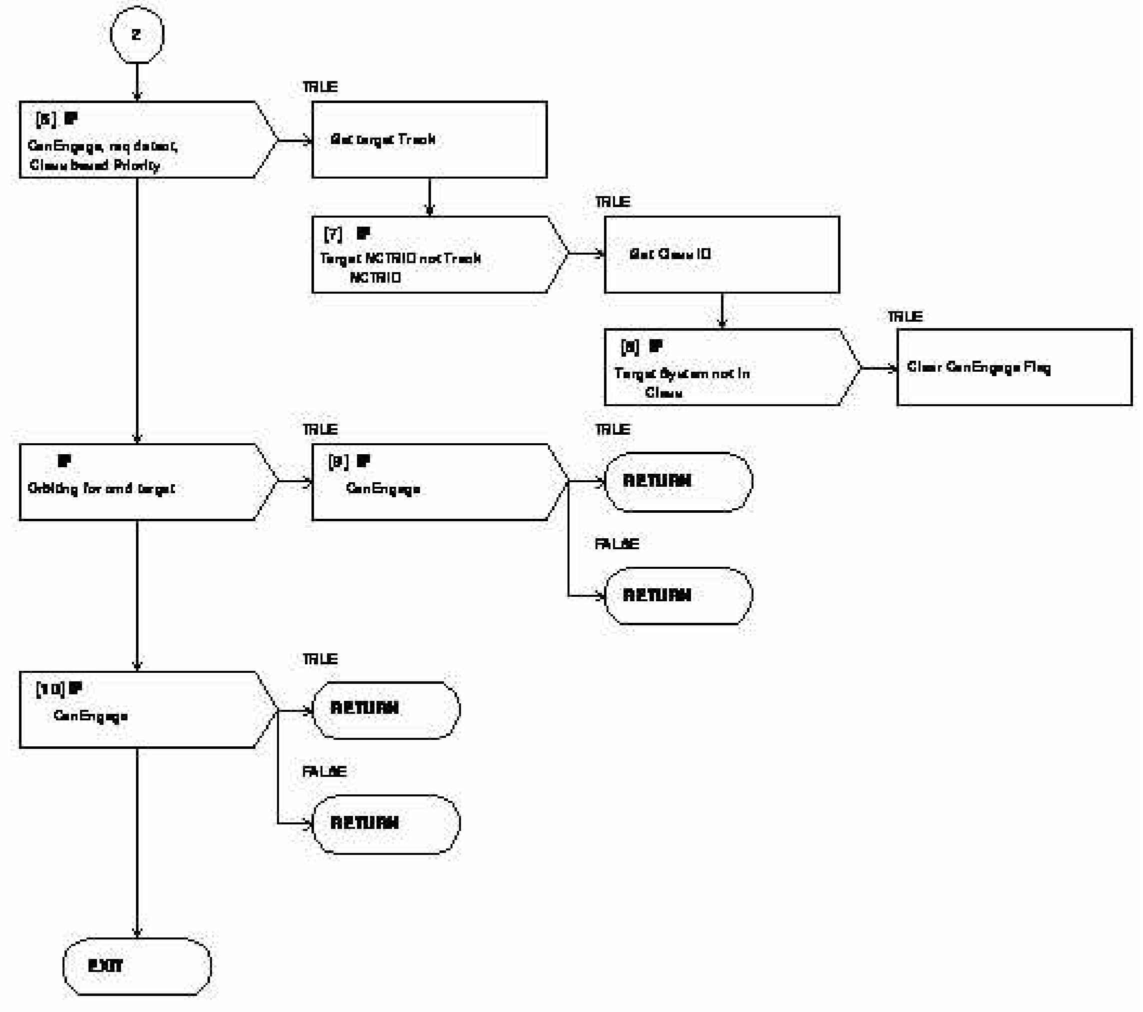

FIGURE 2.11-38. Flow Diagram for ProcessCmdTarget. (Contd.)

Block 1. If the platform has transitioned to normal flight mode upon ending the engagement, return a status indicating “Exit AGPhase1”, else, return a status indicating “Reschedule AGPhase1 at its repeat time”.

Block 2. A check is made to determine if any other target of this type has been flagged as engaged, or the target is flagged as ignore, or the target is flagged as unengageable.

Block 3. If the platform has transitioned to normal flight mode upon ending the engagement, return a status indicating “Exit AGPhase1”, else, return a status indicating “Reschedule AGPhase1 at its repeat time”.

Block 4. If the target is flagged as ignore, or the target is flagged as unengageable, a status is returned indicating “Continue to next target on list”.

Block 5. If the ARM launch conditions have been met for target. a status is returned indicating “Evaluate a weapon for the target”.

Block 6. A check is made to determine if AGAttacker can engage the target, and the target requires detection and class based prioritization is indicated.

Block 7. A check is made to determine if the target record and the track record related to the target have differing target recognition index.

Block 8. A check is made to determine if the target system type is contained in class associated with the track for the target.

Block 9. A check is made to determine if AGAttacker can still engage target. If this is true, a status is returned indicating "Evaluate a weapon for the target", else, a status is returned indicating "Continue to next target on list".

Block 10. A check is made to determine if AGAttacker can still engage target. If this is true, a status is returned indicating "Evaluate a weapon for the target", else, a status is returned indicating "Continue to next target on list".

2.11.4 Assumptions and Limitations

Functional Element	Assumptions	Conditions of Applicability
Battle Management and Message Processing	Note: Battle Management decisions are modeled as a series of phases of operations • The execution of a decision phase is assumed to represent the decisions made since the phase was last scheduled for execution. • The knowledge of engagement activities of other platforms is assumed to be available only through messages received over the communications networks.	• Applies to all rulesets • Applies to all rulesets
Surface-to-Air systems Flexible SAM	• Local sensor are assumed to be cued to a track when a Commanded Assignment message containing a sufficient data quality rating is received at the sensor. • The engagement throughput loading of a SAM system is assumed to be manageable as a function of the number of engagements/assignments over an interval of time. • A target that is known (through received messages) to be engaged by another platform is assumed to be unavailable for engagement. Intentional dual engagements are modeled (time to last launch overrides that allow last shot opportunities by lower tier systems have been added to version 5.00).	• Applies to all tracks in the target select phase of the ruleset • Applies to version 4.01

Functional Element	Assumptions	Conditions of Applicability
Surface-to-Air systems Flexible SAM (Contd)	• SAM systems are incapable of engaging ground targets	
	• SAM systems generate CANTCO messages if they are unable to respond to an engagement command within a specified period of time.	
	• SAM systems assume that a collocated or local sensor with sufficient quality must be tracking a target in order to launch a weapon against it. (Version 5.00 includes the capability to launch weapons using information external or remote sensors with sufficient quality to support engagements in a coordinated manner)	• Applies to version 4.01
	• SAM systems are assumed to require a local impact point prediction before they can evaluate ballistic missiles as threats.	
	• A SAM is assumed to have engagement authority when it is operating in autonomous mode. User inputs determine whether a SAM will attempt engagements in each mode.	
	• An ABT (aircraft or cruise missile) that is receding from an asset is assumed to be non threatening to the asset	• Such targets can be engaged if the SAM is operating in a zone or area defense mode or if the target is approaching another asset.
Decision Making Elements	• Track Establishment does not rely on sensor measurements after the first detection (the M out of N modeling allows intermittent returns).	• This limits the ability to represent the difficulty in establishing track on Low Observable (LO) aircraft.
	• Ground based platforms (such as SAMs) cannot launch another platform for a detailed representation of surface launched cruise missiles or other smart munitions.	